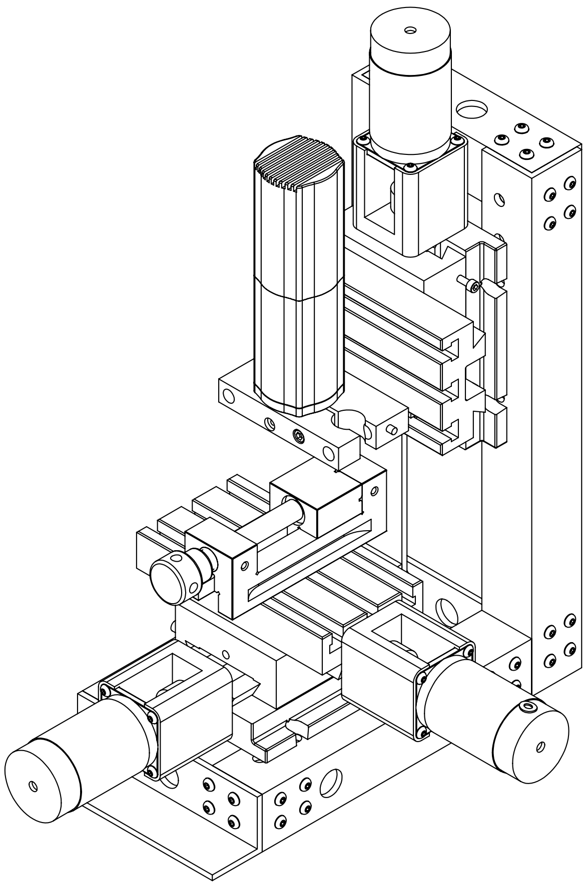
UiO : Department of Physics
University of Oslo

# Self-Improving CNC Milling Machine



Master's Thesis

Torjus Spilling

2014

THESIS FOR THE DEGREE OF MASTER OF SCIENCE

# Self-Improving CNC Milling Machine

*Author:*
Torjus SPILLING

*Primary supervisor:*
Assoc. Prof. Mats HØVIN

*Secondary supervisor:*
Assoc. Prof. Kyrre GLETTE

University of Oslo

2014

www.duo.uio.no

Printed by Reprosentralen, University of Oslo

**Abstract**

This thesis is a study of the ability of a CNC milling machine to create parts for itself, and an evaluation of whether or not the machine is able to improve itself by creating new machine parts.

This will be explored by using off-the-shelf parts to build an initial machine, using 3D printing/rapid prototyping to create any special parts needed for the initial build. After an initial working machine is completed, the design of the machine parts will be adjusted so that the machine can start producing its own machine parts, replacing the 3D printed parts. It is desirable for the finished machine to have as many self-created parts as possible.

As a physical machine will be built, the steps needed to create the machine will be detailed, so that the machine can be replicated if desired. The finished machine will not be autonomous, but will rely on a human operator to design the parts and operate the machine.

# Acknowledgements

I would like to sincerely thank my primary supervisor Mats Høvin for his support during the work of this thesis. He has been very helpful, and has always been accommodating when I've needed building materials or access to machine tools or other equipment. He has been a great inspiration with all his own projects, and has taught me much about prototyping.

I would also like to thank my secondary supervisor Kyrre Glette for his helpful pointers and tips on the writing of the thesis.

Thank you to Yngve Hafting for assisting practical work and making sure I kept myself safe while doing different machine work.

Thank you to Jim Tørresen for including me in the research group for robotics and intelligent systems and for providing me with a research assistant position during my time as a student. The elevated machine access that comes with the position has been much appreciated.

Lastly, I would like to thank Gunhild Karlsen, my beautiful future wife, for helping read through my work and for giving helpful pointers. Thank you for putting up with my long rants about CNC machines and also my occasional frustrations during the work.

# Contents

# List of Figures

All photographs and drawings by the author unless otherwise specified.

# List of Tables

# Glossary

**CNC** Computer Numerical Control

**CAD** Computer-Aided Design

**CAM** Computer-Aided Manufacturing

**DIY** Do It Yourself

**DRO** Digital Read-Out

**FDM** Fused Deposition Modeling

**FEA** Finite Element Analysis

**FEM** Finite Element Method

**FFF** Fused Filament Fabrication

**G-code** A language for describing CNC toolpaths

**HSM** High Speed Machining

**IDE** Integrated Development Environment

**ISO** International Organization for Standardisation. As the three official languages of the organization are English, French and Russian, the abbreviation would be different in all three languages. Therefore, the abbreviation agreed on, ISO, comes from the greek word *isos* meaning equal (*ISO website* n.d.).

**MDF** Medium Density Fiberboard

**PCB** Printed Circuit Board

**PID controller** Proportional-Integral-Derivative controller

**ROBIN** Robotics and Intelligent systems research group at the University of Oslo. The research group where this thesis has been written.

**SVG** Scalable Vector Image

# Chapter 1

# Introduction

## 1.1 Motivation

This thesis will explore the ability of a CNC milling machine to create parts for itself. CNC is short for computer numerical control, and a CNC milling machine is a milling machine where the individual axes of the machine are controlled by a computer through a motor. The motivation for choosing this particular topic for the thesis lies in the challenge of building the actual CNC milling machine and the ability to create complex and repeatable parts with it. The fascination lies in the joy of creating something tangible, making a machine which further pushes the boundaries of creativity. The opportunity to be able to create a physical machine will also serve as an enjoyable way to gain experience with working with different materials, making use of the different machine tools at the University of Oslo's disposal.

The hands-on experience is a reason as important as any for choosing this particular topic. Teaching oneself something by actually touching, measuring, and adjusting things to make them work is immensely rewarding and really builds an understanding of how things interact mechanically.

## 1.2 Goals of the thesis

The goal of the thesis is to build a three-axis computer controlled milling machine. The machine will be built at the robotics and intelligent systems (ROBIN) research group at the University of Oslo, and it will reside there after completion. The intention is that the machine, when completed, may serve as a non-expensive CNC teaching aid when teaching new students to use CNC machines. Another goal is that the machine should be accurate enough to produce parts for robotic projects at the research group. It will be an addition to the machine park at the research group. For the machine to have any hope of being used, it needs to be made relatively intuitive and easily operable. The work will be done from the viewpoint of a tinkerer or a hobbyist, frequently utilizing materials currently at hand to solve different problems. It also means trying to keep the total cost of materials used in the construction of the machine as low as possible.

The basis of the machine will be two cast iron X/Y-coordinate tables and three powerful stepper motors, one for each of the machine's axes. The coordinate tables will need to be fixed to a mounting frame so that they are located at a right angle in relation to each other. The stepper motors will need to be fixed to the coordinate tables, replacing the manual adjusting wheels, and these will need to be controlled with electronics.

When a functional machine is built, areas for improvement will be discovered, and the machine will be put to work to create better parts and solutions for itself. The aim will be for the machine to create as many parts for itself as possible, as this is the method of self-improvement the thesis will focus on.

The most important feature of the CNC milling machine is its precision, therefore improving the precision will be the main goal when improving the machine. This will be done by ensuring rigidity in the machine, and minimizing the backlash in the mechanisms controlling the axes. Other possible areas for improvement are:

- Operating speed of the machine.
- Size of the work area. A bigger work area allows creating bigger parts on the machine.
- Minimizing physical machine footprint without sacrificing work area.
- Ease of use. Increases the chances of the machine actually being put to use to create parts after it has been constructed.
- Cost of construction.
- Elegance and aesthetics. Although the appearance of a CNC milling machine does not affect the output, it may be a factor for a DIY (do-it-yourself) user who may not have space to store the machine out of sight.

As the milling machine evolves, the machine is intended to become better, and the desired end result is a machine which is able to work with stock materials such as plastic, wood and even light metals like aluminum and bronze. Working with metals will give the CNC machine an advantage over 3D printers. Metals are often materials more suitable for making durable parts when compared to the plastic used by 3D printers.

To commence studying of the CNC milling machine, the following specific problems are stated:

1. How can a self-improving CNC milling machine be created?
2. Which components of a CNC machine are the ones which are most suitable for improvement?

## 1.3 Outline

The thesis is divided into eight chapters: introduction, background, tools and engineering processes, experiments, implementation, improvements, resulting machine, and discussion.

The background chapter gives an overview of the theory behind the work of this thesis, and presents previous projects which have served as inspiration for this work. A selection of existing CNC machines aimed at hobbyist use are discussed, and their abilities for self-improvement are evaluated. An overview is given over some of the ways CNC machine precision is measured.

The tools and engineering processes chapter gives an overview of what methods will be used to build the CNC machine of this thesis. Software tools used to design the machine and to create CNC machine toolpaths are presented. Hardware tools used to create mechanical parts and electronics components that will be used in the main CNC machine build are introduced. The electronics used to drive the CNC machine will also be discussed.

In the experiments chapter, two existing small CNC machines built at the research group are presented, and these are used to gain experience with working with CNC machines. Some setup is necessary before being able to use the machines, and the setup process is described in this chapter. Working with the smaller machines provides experience which will prove useful when moving on to building a bigger CNC machine. Results from the small CNC machine experiments will be presented.

The implementation chapter details the process of building the main CNC milling machine from scratch. Design choices are discussed and explanations are offered for the different methods or materials that have been used. Mechanical hardware used to build the CNC machine is presented, and custom electronics necessary to run the CNC machine is discussed.

In the improvements chapter, the CNC milling machine has been built, and different modifications are made to the machine. The goal of the modifications is to improve different aspects of the built CNC milling machine. Many of the modifications are performed on the physical machine, but some hypothetical improvements are also proposed towards the end of the chapter.

The resulting machine chapter showcases the built CNC milling machine after the improvements have been performed. Technical specifications, cost and self-improvement results are presented.

The final chapter contains a general discussion of the work done in the thesis, gives a conclusion, and offers suggestions for future work.

# Chapter 2

# Background

This chapter will go through the theory which forms the foundation for the thesis. Some basic concepts in manufacturing are explained, and the idea of self-replication and self-improvement is introduced, together with the self-replication experiment the RepRap project. A selection of existing CNC machines aimed at the hobbyist market is presented, pointing out some of the features of these, along with an evaluation of the abilities of these machines to perform self-improvement by making parts for themselves. The selection of the studied CNC machines has been made because they are all targeted towards hobbyist end users who need to assemble the machine themselves. Towards the end of the chapter some of the metrology involved in determining CNC machine precision is presented.

## 2.1 Additive and subtractive manufacturing

There has been a lot of excitement about ever cheaper 3D printers hitting the market. 3D printing is an additive way of manufacturing where melted material, often plastic, is used to build up a model. 3D printers are able to create complex three dimensional structures easily, by building up a model layer by layer. They are also easy to use, requiring a user to merely be able to create a 3D-design in some CAD (computer-aided design) software. On the other hand, 3D printers are in general limited to working with plastic, and although they are getting more and more accurate, there is a limit to how accurate the finish on parts created with a 3D printer can become. Plastic may also not be a durable or rigid enough material for production use.

In subtractive manufacturing, the process starts with a piece of stock of some material, for example hard plastic, wood, aluminum, and with stronger machines, steel. CNC milling machines are the subtractive counterparts of 3D printers. They tend to be heavier and bulkier than equivalent 3D printers. This is because removing material from stock demands more power from the axis motors, and more rigidity from the support structure. There is also a higher learning threshold to overcome before using a CNC mill, as the user is not only required to be able to model their part in CAD software, but also needs to be able to decompose their model into the machine toolpaths used to cut the workpiece, usually using a separate CAM (computer-aided manufacturing) software package.

Subtractive manufacturing such as CNC machining will generate waste in the form of wood, plastic or metal chips that are removed in the machining process. This waste will have to be disposed of in a suitable manner during the machining process to prevent clogging up the machining tool. Clogging causes excessive wear on the tool bit and may even cause the tool bit to break, which in turn could lead to a ruined part.

Care needs to be taken to ensure human safety when using a CNC machine. There are great forces at work in the machine, something which could potentially pose a high risk to the operator of the machine, at least if compared to operating a 3D printer. Wood or metal chips or even shards from broken tool bits can potentially be flung from the machine. A shield or an enclosure will usually be a part of the machine, and an emergency stop button is essential in order to quickly be able to kill power to the machine if something goes wrong. The CNC machine will also need to be used under supervision.

Despite the described obstacles associated with the use of CNC milling machines, this kind of machine is used if a precision part is needed. CNC milling machines outperform 3D printers when it comes to precision.

## 2.2 Self-replication and self-improvement

A self-replicating machine is a machine that is capable of reproducing itself autonomously from raw materials. John von Neumann pioneered the concept by imagining a conceptual machine with unlimited access to energy and a supply of machine parts in a warehouse (Neumann, 1966). The machine would also have assembly instructions for how to create itself. The idea was that the machine could then reproduce a replica of itself using a robotic arm. The machine would pick up a machine part, examine it, and use it if this was the currently needed part. Otherwise, if it was determined that this part was not needed at the moment, the machine would return the part, and try another part. After all the parts were assembled into the new machine, the original machine would copy its instruction set into the new machine's memory.

Even before the emergence of numerical control, machine tools such as lathes or milling machines have been considered a unique form of machines. As long time machinist and machine handbook author Fred H. Colvin points out at the start of his autobiography: “For machine tools are the *only class of machines that can reproduce themselves.*” (Colvin et al., 1947, p. 6). In this case it is assumed that a human operator will be operating the machine, taking measurements and creating the individual machine parts. The machines are not autonomous.

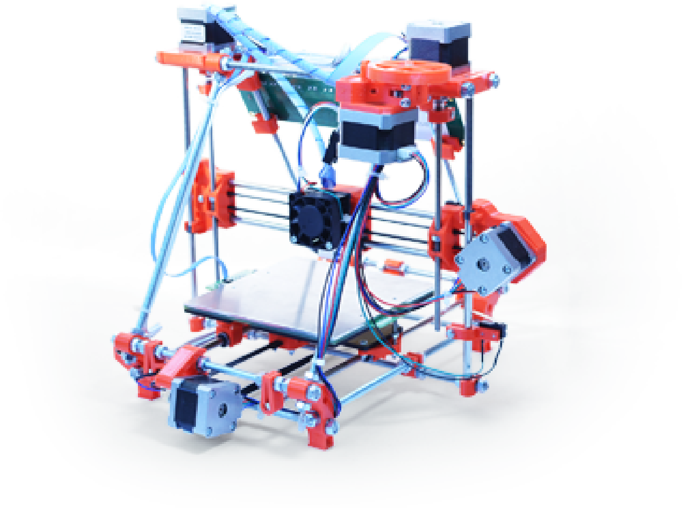


**Fig. 2.1** RepRap version 3 “Huxley”. (Photo: www.reprappro.com)

## 2.3 The RepRap project

The RepRap project (Jones et al., 2011) has explored the concept of self-replication. RepRap is an abbreviation of Replicating Rapid Prototyper, and is a fused-filament fabrication 3D printer which uses melted plastic to print object. It is designed to be as simple as possible to build, using 3D printed parts and commonly available hardware. The design of the machine is distributed freely under the GNU General Public License (Stallman, 2007). What makes the RepRap special, is that it is to a large extent capable of manufacturing the parts needed to create a copy of itself. By its own definition, the RepRap is a kinematic assisted self-replication and self-manufacturing machine. The machine is kinematic since it is a physical machine as opposed to just a software model of a machine. It is assisted because it relies on the assistance of a human operator to assemble the individual machine parts to make a machine.

The human operator is thus a part of the self-replication process, assisting the machine for the mutual benefit of the RepRap machine and the operator. The RepRap is given the ability to reproduce with the help of the human operator, and the human operator is given the opportunity to manufacture useful objects whenever the RepRap is not creating parts for replicating itself. This codependence between the machine and the human operator emulates mutual benefit relationships found in nature, such as a bee who gets its nectar from flowers and in return spreads the pollen of those flowers (Bowyer, 2014).

The RepRap has been successful in that it has contributed to a rise in the popularity of 3D printers, and getting more people interested in the technology. Due to the open nature of the project, many have contributed to the evolution of the RepRap. The RepRap has spawned many spin-off 3D printers and has been the inspiration of a number of commercial companies who create 3D printers.

## 2.4 Existing open-source CNC milling machines

As CNC machining is a mature technology, there are a number of CNC milling machines already available. Some of these are open-source and their plans have been made available online. In this section, some of these CNC machines are described, and their features detailed. The different machines are evaluated and their perceived ability for self-improvement by creating parts for themselves are discussed.

### 2.4.1 DIY LIL CNC

Taylor Hokanson and Chris Reilly of School of the Art Institute of Chicago are the research team behind the open source CNC machine which they have dubbed DIY LIL CNC (Hokanson et al., 2009). The design of the machine was based on an existing design found online, a CNC machine by someone who calls himself *oomlout* online (oomlout, 2007) (*Instructables website* n.d.). The oomlout CNC machine used barrel nuts inserted into the ends of MDF (medium density fiberboard) boards, which allowed boards to be joined together by inserting machine screws into the barrel nuts.

Making holes for the barrel nuts in the source design involved a lot of drilling into the ends of boards, which is time consuming and tedious, therefore the DIY LIL CNC instead opts for a solution where plywood boards may be laser cut with room for standard nuts without any drilling into board ends. This made the individual structural elements easier to manufacture.

The DIY LIL CNC does a good job of creating a low-cost CNC machine, but being constructed out of wood limits the rigidity of the machine, and also limits which materials can be milled with the machine. A second version of the machine has been created called DIY LIL CNC v2.0 “Big Shoulders”, but while measures have been taken to increase the rigidity of the machine, the machine is still based on a wooden structure.

The DIY LIL CNC is a gantry-style CNC machine, also called a CNC router, meaning that the whole gantry where the spindle is mounted is moved when the machine is operating. The workpiece is held stationary. This style of machine is the counterpart of a CNC mill, where the spindle is only moved up and down in the Z-direction, and the X- and Y-motion is taken care of by a moving work table, moving the workpiece with it. A CNC mill often has a larger clearance in the up and down Z-direction, which makes it easier to use a vise to hold the workpiece. A CNC router, on the other hand, often works with sheets or plates of material, or other relatively flat workpieces.

The DIY LIL CNC is not provided as a kit for purchase. All cutting of the structural components and sourcing of hardware must be done by the end-user. The creators of the DIY LIL CNC merely provide the plans for how to

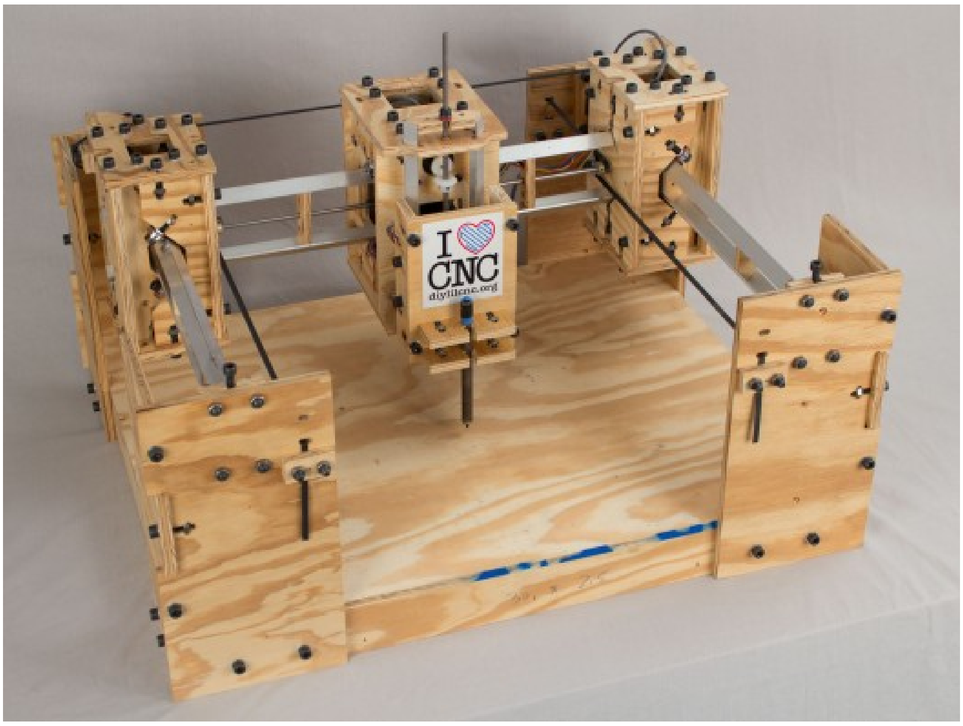



**Fig. 2.2** DIY LIL CNC v2.0 “Big Shoulders” (Photo: www.diylilcnc.org)

make the machine and a bill of materials guiding the user to where to buy components (Hokanson et al., 2011).

It looks like the DIY LIL CNC could be put together from simple parts and materials, and the wooden parts making up the machine could be cut quickly and accurately with the help of a laser cutter. Although the machine could be created from inexpensive and readily available materials, the design looks to be a bit complicated, at least when compared to the Shapeoko which will be discussed later. The relative complexity of the design, and the fact that a laser cutter is not among the machines that the research group for robotics and intelligent systems possesses, means that this design will not be used as the base for the CNC machine that will be built during the work with this thesis.

### 2.4.2 Project Shapeoko

Shapeoko is a project aiming to enable people to easily assemble or scratch-build an easy-to-use desktop CNC machine. The Shapeoko is a CNC router designed by Edward Ford (*Project Shapeoko website* 2011) where the gantry holding the milling tool moves, while the worktable is stationary. Its design, including all parts lists and drawings, are all licensed under the Creative Commons license, meaning that it is freely distributable as long as the creator is credited. The Shapeoko CNC machine utilizes freely available open-source software in its software toolchain. In addition a wiki has been set up

**Fig. 2.3** Shapeoko 2, released Oct. 2013. (Photo: www.inventables.com)

allowing users to exchange ideas and knowledge (*Shapeoko Wiki* 2011).

By making the machine design simple, releasing the design as open hardware, and making the components easily available online, the creators of the Shapeoko hope to popularize desktop CNC fabrication much like desktop 3D printing has become popular in later years. The machine makes use of the MakerSlide for its linear motion bearings. This is a rigid aluminum profile, designed by Barton Dring, created for hobbyists wishing to create CNC machines, and it utilizes a V wheel running on a V rail (Dring, 2011). The stepper motors of the Shapeoko are controlled by an Arduino preloaded with open-source CNC-controlling firmware Grbl. Grbl was initially written in 2009 by the Norwegian Simen Svale Skogsrud, and since 2011 led and developed by Sonny Jeon (Skogsrud, n.d.). Since the whole weight of the gantry is moved during the operation of the machine, two stepper motors working in tandem power the Y-axis motion to divide the labor of moving the gantry. The two stepper motors are mounted at opposite sides of the gantry, ensuring that there is as little angular deformation in the gantry as possible during operation.

The Shapeoko 2 is available for purchase as three different ready-to-assembled kits. The mechanical kit includes only the skeleton of the machine, leaving the user to get hold of stepper motors, spindle, belts, pulleys, and mounting hardware on his own. The full kit includes everything needed to get the machine up and running, including stepper motors and a spindle. A third kit called the works includes everything from the full kit in addition to upgrades such as an improved spindle, a wasteboard, and an electronics enclosure.

Seeing a need to provide a simple means of producing G-code, Edward Ford has also revived an abandoned online CAM software package he discovered, which he calls MakerCAM. It is capable of creating toolpaths for pocketing operations, profiling operations, and drilling holes. These CAM operations are described in more detail later, in section 3.8.1. MakerCAM can generate toolpaths from an imported SVG (scalable vector image) image, or the toolpaths may be created directly on the MakerCAM website (*MakerCAM website* n.d.).

The Shapeoko seems to be popular among hobbyists because of the simple design. The design uses as few components as possible to keep the cost down. In addition, the size of the machine is easily increased by using longer MakerSlides for the X- and Y-axis rails. Because of the popularity of the machine, there is also a lot of documentation online about various modifications and improvements that users have made to it.

This seems to imply that the machine would be a good candidate to use as a self-improving machine. The reason that a Shapeoko has not been created for work with the thesis, is that the default design of the machine looks like it may have problems with rigidity, and the CNC machine created during the work with this thesis should be rigid enough to machine aluminum relatively easily.

### 2.4.3 The Creation Station

The Creation Station is a project funded through Kickstarter, and it aims to create a desktop CNC machine for hobbyists (*The Creation Station CNC Router website* n.d.). The machine is an attempt to overcome some of the limitations of other hobbyist CNC machines, one of which is the small build envelope often found in these machines. The Creation Station has a work area of 609 mm × 609 mm × 127 mm, and this is by far the largest work area of machines described here. Another thing that separates the Creation Station CNC machine from other hobbyist machines is the use of ball screws for the separate axes of the machine. This feature reduces mechanical backlash in the machine, which in turn leads to higher precision in the milled parts created with the machine. During operation, the gantry of the Creation Station is stationary with the tool moving in the X- and Z-axes on the gantry. The Y-axis movement is taken care of by moving the table of the machine.

The Creation Station is offered as three different kits. The starter pack only includes the machine building kit, and does not come with a spindle. The maker bundle comes with the machine kit, a spindle and software Vectric Cut2D software for 2D cutting. The creator bundle comes with a spindle and the more feature-rich software VCarvePro.

Although the machine is actively marketed as an open source machine, the plans for the machine are difficult to track down. This may be due to the fact that the company making the machine, Innovation Squared LLC, is at the time of writing just finishing their initial order after a successful Kickstarter

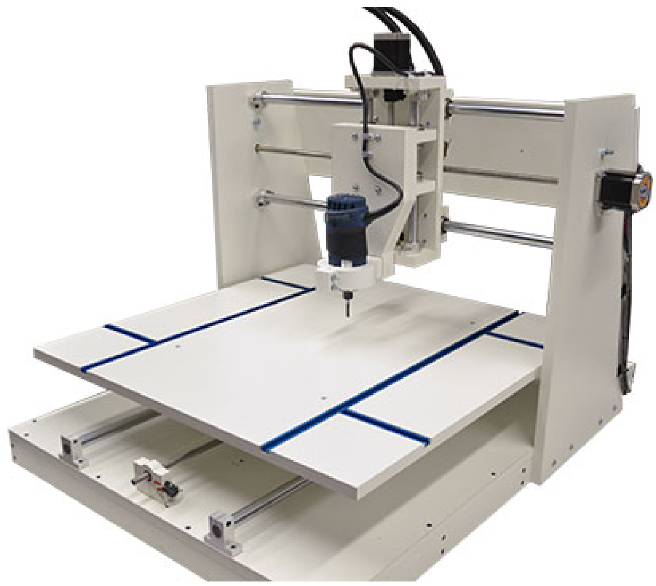


**Fig. 2.4** The Creation Station (Photo: www.creationstationcnc.com)

crowdfunding campaign (Innovation Squared LLC, 2014). The plans may be released after their initial order has shipped.

The Creation Station looks like a solid machine, and seems to be aimed towards hobbyist users who are more into creating stuff with their CNC machine than to tinker with and upgrade the machine itself. The use of ball screws in the machine seems to imply that this is a precise machine, and it looks like the machine is powerful and sturdy enough to handle a big spindle and still stay rigid.

One reason for not using this machine with the thesis work is that it seems like the creators of the machine are not too keen on shipping the machine internationally. It further seems like the machine is not quite ready to be shipped yet anyway, as the website gives an invitation to place pre-orders for the machine. The other reason for not using this machine is the fact that the plans for the machine do not seem to be published yet. If the creators of the machine intend to release the plans, it does not seem like they intend to do so before having a fully developed machine that is ready to be shipped.

### 2.4.4 Lobo CNC milling machine

The Lobo CNC is an open-source project where the machine table moves in the X- and Y-directions while the milling tool is only moved in the Z-direction. The Lobo CNC was designed in this way to be able to have a larger Z-axis range of motion, and to allow for easy use of a vise to clamp workpieces to. Its advantage over a moving-gantry style design is that objects of irregular shapes are more easily milled than in a gantry CNC machine where flat stock materials are easier to work with. This CNC mill design also allows for easier workpiece clamping, as there is room for using a machine vise with the

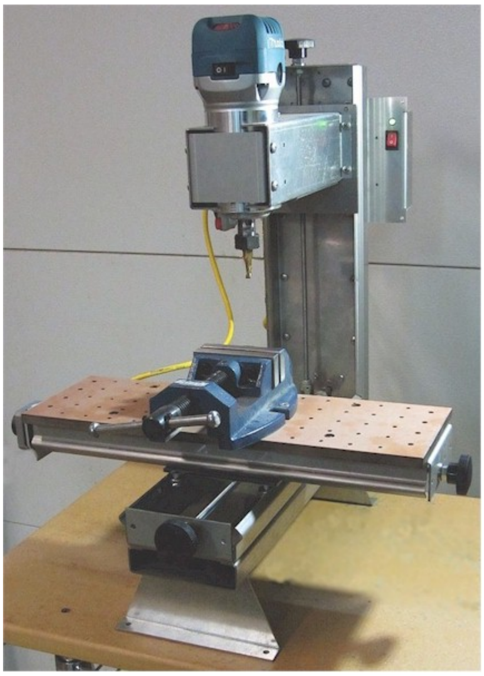


**Fig. 2.5** Lobo CNC milling machine. (Photo: www.jrkerr.com/lobocnc)

machine.

The creator of the Lobo CNC machine, Jeff Kerr, has also devised a closed-loop control system for stepper motors for use in the machine (Kerr, 2013). This means that rather than driving the motors step by step, they can be run continuously like brushless servo motors and using a PID (proportional-integral-derivative) control system to position the motor exactly. This means the machine operator does not need to worry about missing steps, which may be the case when using stepper motors on their own, because positioning errors will be taken up and corrected by the feedback control system.

An advantage of the servo motor approach to driving the axis motors is that the motor control may be switched off to turn the axes by hand, and the control system will still be able to keep track of the position of the axes, remembering the homing information. This allows for quick, easy and intuitive repositioning of the axes. The machine may also be used as a manual milling machine with a digital readout (DRO).

The machine structure is made out of laser-cut steel sheets which gives the machine its necessary stability. The Lobo CNC milling machine is shown in Fig. 2.5. Although the CNC milling machine is an open-source machine, the firmware of the individual microcontrollers in the electronics remains closed source, as the designer of the machine, does not own the rights to all of the firmware in the controllers, and as such he can not release the source code.

The Lobo CNC is available for purchase from the creator's website (Kerr, n.d.). It is offered as three different kits, one including the mild steel sheet metal parts which make up the mechanical structure of the machine, another for the stepper controller board and the three stepper motors with rotary encoders used with the machine, and a third kit for the off-the-shelf hardware which would be expensive in small quantities or would have been a nuisance to buy from separate vendors.

The plans for the Lobo CNC machine are easy to locate, but the creation of the individual parts of the machine made out of mild steel sheets require accurate folding of the sheets to create usable parts. The parts for sale by the creator of the machine have been laser cut and folded by a CNC folding machine. These are specialized machines that may not be available to a hobbyist, and they are not available to the author either, which is a reason that the design of the machine has not been used for the CNC machine built during the thesis work. The design of the Lobo CNC is a good one, though. The machine to be built will be somewhat similar in concept, being created as a CNC mill rather than a CNC router. The machine will however have to have a different design that allows it to be created with the equipment available at the ROBIN research group.

### 2.4.5 CNC machine comparison

To compare the discussed CNC machines to each other Table 2.1 has been compiled. None of these machines come pre-assembled, to reduce on cost, although all but the DIY LIL CNC can be purchased as ready-to-assemble kits where the structural components have been created and all the hardware to mount it all together has been bundled with the kit.

The price comparison in Table 2.1 is based on the price for a machine with all components needed to get it working, including a spindle and the axis-driving motors. For the DIY LIL CNC this price is a ballpark figure, as the machine is not sold as a kit. The price of the Shapeoko 2 is for the full kit, with the European 230 V power option. The price shown for the Creation Station is for the maker bundle. The price of the Lobo CNC is based on the estimated price given on the Lobo CNC website. This includes $50 for an adequate spindle, but the website also lists more powerful and more expensive alternative spindles which will fit the machine.

The prices are listed in U.S. dollars, as this is the currency the machines are offered in. None of the prices include shipping fees or import taxes. The prices have also been converted to Norwegian kroner, with a conversion rate where 1 USD equals 6.8 NOK. This is to give some basis for comparison with the machine which will be built, and which will be discussed later in the report.

## 2.5 Open source and the maker movement

Both the RepRap and many of the various home CNC projects have made the plans for their machines freely available for download. In many cases this does not interfere with the commercial aspects of the projects, as the creators

**Table 2.1** Open source CNC machine comparison

| | DIY LIL CNC v2 | Shapeoko 2 | Creation Station | Lobo CNC |
|---|---|---|---|---|
| Axis motors | NEMA 17 steppers | NEMA 17 steppers (2 for X-axis) | NEMA 23 steppers | NEMA 17 steppers run as servos |
| Spindle | Dremel | Dremel size rotary tool | Bosch Colt 750 W | Grizzly H7790 440 W |
| Size (mm) | 809 × 654 × 465 | 550 × 510 footprint | 1118 × 838 × 813 | 381 × 102 work table |
| Work area (mm) | 310 × 340 × 110 | 298 × 301 × 54 | 609 × 609 × 127 | 254 × 203 × 178 |
| Price (USD) | ~700 | 695 | 1149 | 878 |
| Price (NOK) | ~4760 | ~4726 | ~7813 | ~5970 |

of the machines may still make a profit from compiling kits that are ready for assembly and selling these to makers that are interested in doing their build as conveniently as possible. These kits may also be compiled from parts made or purchased in bulk, increasing the profit margin of the kits.

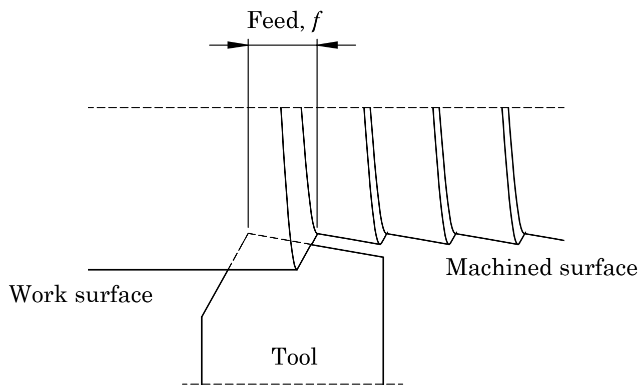
Releasing the design of a machine for the hobbyist market as open source seems to be a good marketing strategy. Many hobbyists seem to prefer open-source designs as having the plans of the machine available simplifies modifications and improvements to the design, which may be some of the attraction with the machine.

Open source fuels the maker movement. Instead of trying to keep the construction of the machines secret the plans are distributed freely, and end users are encouraged to build, modify, and share their modifications back into the open source community. In this way the creators of the projects are fed with new ideas on how to improve their machines, which is a benefit for all parties involved. Profit is not the core incentive of the maker movement (Lipson et al., 2013, p. 50), but makers are instead driven by a curiosity and a problem-solving desire.

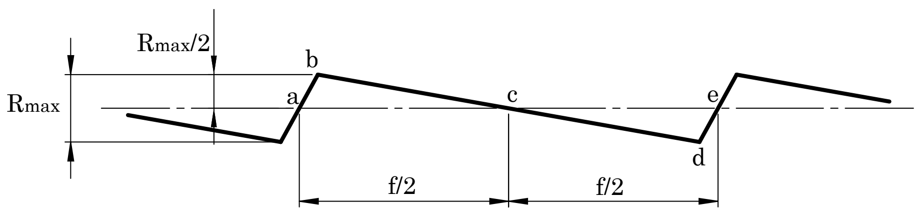
Ideally the design for the machine which will be built during the work of this thesis would be open sourced from the start and published at a suitable site online. The motivation for this would be to be able to contribute something to the community, in the hope that someone would find it useful and be able to make use of the information. However, due to the inherent time constraints that apply when writing a master's thesis, the design files have not been made available. This is only because of time prioritization and not because of any desire to make it difficult to recreate the design. The design may be published online at a later time, after the master's thesis has been completed.

## 2.6 Machining metrology

There are many ways of measuring the precision of a CNC machine, and a few of the concepts and measuring techniques used to do this are described




**(a)** Lathe tool machining a rotating workpiece from right to left.




**(b)** Detail of machined surface

**Fig. 2.6** Surface roughness calculation. Illustrations based on an image from the website CYBERMAN (Sutherland, 2002).

below. The aim of this is to give an impression of how different factors affect the precision of a CNC machine.

### 2.6.1 Average surface roughness, $R_a$

The average surface roughness is a metric for how smooth a machined surface is. It gives a good general description of the height variations in the surface. To calculate the average surface roughness, a mean line parallel to the general surface direction is found, dividing the surface in such a way that the sum of the cross-sectional areas formed above the mean line is equal to the sum of the cross-sectional areas formed below the mean line. The average surface roughness is given by the sum of the absolute values of these areas, divided by a sampling length. Using the naming from Fig. 2.6, the following equation defines the average surface roughness:

$$R_a = \frac{|A_{abc}| + |A_{cde}|}{f} \tag{2.1}$$

Here, $A_{abc}$ and $A_{cde}$ denote the areas above and below the mean line, respectively. $R_a$ is the average surface roughness. Looking at Fig. 2.6b one can see that removing more material in a pass will result in a larger maximum roughness value $R_{max}$. This means that the average surface roughness value will increase if more material is removed. When seeking to obtain a good surface finish quality, just a small amount of material should be removed. This is what is referred to as a finishing pass. The opposite type of operation is the roughing pass. In a roughing pass the focus is to remove as much material as possible as quickly as possible. Roughing passes are usually followed by a finishing pass.

In can be further determined that when the feed is lower, this gives a lower height of the triangles $A_{abc}$ and $A_{cde}$, as points $b$ and point $d$ lie closer to the mean line. A lower feed is obtained by moving the tool at a slower rate or by rotating the work at a faster rate. This means that a lower feed speed or a higher work rotation speed will also decrease the average surface roughness $R_a$. A finishing pass where a smooth surface finish is desired should be made with a lower feed speed in addition to removing just a small amount of material.

Although Fig. 2.6 shows a workpiece being machined on a rotating lathe, the same concepts apply to a milling machine where the workpiece is kept stationary while a milling tool is rotated and moved through the workpiece.

### 2.6.2 Tool deflection

As a milling tool is moved into a workpiece to start the cutting, the milling tool will flex a certain amount. The distance between the desired tool tip placement and the real world position of the tool tip is called the tool deflection. Tool material and work material are factors that influence the tool deflection. In addition feed speed, rotational speed and depth of cut have a significant influence on how much the tool deflects.

Instead of always counteracting tool deflection, manufacturing is often divided into roughing operations and finishing operations. During the roughing operations, a large volume of material is removed. Some material is left in place, and therefore tool deflection can be accepted, and feed speeds can be high.

After a rough approximation is achieved with the roughing operations, finishing operations are run to get a proper surface finish. During these operations, the workpiece is milled to accurate dimensions by a fast-spinning tool with a low feed speed. As there is less material to remove after the roughing phase, tool deflection is low, and a good surface finish is achieved.

### 2.6.3 Ballbar

A ballbar is a device which is used to test the accuracy of a CNC machine. It consists of a digital scale which at one end is attached to the spindle of the CNC machine and at the other end is attached to a fixed point on the work table. The device is called a ballbar as the digital scale has a ball at either

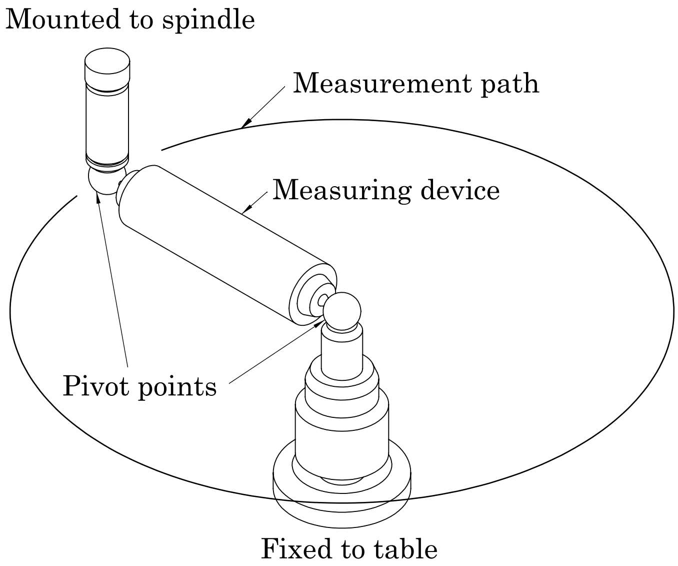



**Fig. 2.7** Ballbar and measurement path.

end. The balls are mounted in magnetic cups on the spindle side and at the fixed point side. When the ballbar has been mounted in place, the CNC machine is programmed to draw out a circle. The ballbar accurately monitors any change in radius as the machine follows the circular path, and the resulting data is fed to a computer where it is recorded. A computer program can then analyze the data to extracts error values for different parameters and to plot the results of the test onto a graph. Values for backlash, reversal spikes, lateral play and cyclic error can all be determined from a ballbar test. A ballbar is shown in Fig. 2.7.

# Chapter 3

# Tools and Engineering Processes

This chapter will detail the tools that will be used to perform the work with the thesis. This includes some of the most important physical tools, but the software tools will also be discussed. Various engineering and design processes will be described. The workings of stepper type motors will be detailed, as these will be the types of motors driving the individual axes in the final machine. This also includes a description of the electronics used to drive the stepper motors. Towards the end of the chapter an overview of computer-aided manufacturing will be given.

## 3.1 Computer-aided design

The design process for a new product will often start with creating a 3D computer model of the object. Creating 3D models of the individual components used to build the final machine is a good aid in visualizing how the components will fit together. This can uncover collisions and misalignments that have been overlooked, which may be difficult to spot before the machine is viewed as a whole. In this way creating a 3D model of the whole machine saves time and possibly working material when the actual building of the CNC machine takes place.

With a 3D model available, a finite element analysis (FEA) can be performed on the different parts of the machine. FEA is the practical application of FEM (finite element method), and in this type of analysis a part is divided into a mesh so that each surface within the mesh can be used for calculations. Dividing the part into a mesh such as this simplifies the calculations, and the calculations are performed on all of the surfaces of the model so that an approximation to how the complete model performs can be found. In this way different analyses can be performed on the part, for example to find out how forces act on a part, which areas of a part are most severely stressed, or how temperature affects a part.

## 3.2 SolidWorks

SolidWorks is a 3D CAD software package produced by Dassault Systèmes SolidWorks Corp. 3D models in SolidWorks are built up in individual operations by adding features, and dimensions are added to the features to fix the sizes of the different elements of the model. All features added to the model are arranged chronologically in a feature tree. This parametric approach makes it easy to modify the design by going back and modifying parameters which were given earlier in the modeling process.

SolidWorks is designed to be a powerful 3D modeling software package which is easy to use. Parts are created and may then be combined into assemblies or sub-assemblies. Technical drawings may then easily be generated from assemblies or parts.

During the thesis work all parts of the CNC milling machine have been modeled in SolidWorks prior to attempting to create them physically. By doing this potential problems are discovered early, such as colliding parts or wrong screw sizes. It is also a way to make sure that connecting parts fit together.

SolidWorks was chosen as the CAD solution for the thesis work as it is a widely used CAD software package. It was also readily available at the University of Oslo.

## 3.3 3D printing

After a 3D model has been created through the use of CAD tools, a physical plastic part may be created with the help of a 3D printer. The 3D model is fed into a slicing program which creates the toolpaths that will tell the plastic extruder head of the 3D printer how to create the part. The toolpaths are then sent to the 3D printer which will set to work creating the part.

Two different 3D printers have been available during the work with this thesis. The two 3D printers are described in the following sections.

### 3.3.1 Objet Connex 500

The Connex 500 3D printer was manufactured by Objet Geometries. Objet Geometries has since merged with the company Stratasys, and the Connex 500 is now sold as part of Stratasys' Objet line. The Connex 500 uses PolyJet technology, which means that the model is built by applying layers of liquid photopolymer which are instantly cured into hard plastic. When creating overhanging parts in a model, a gel is deposited as a support material.

The resulting finish on the model created by the Connex 500 3D printer is very good, with a very smooth surface. However, while the model stays true to its intended form right after it has been printed and cleaned up, the material has been known to warp with time, making it difficult to create precision parts with this machine. This is a big problem when creating parts for a precision machine.

### 3.3.2 Stratasys Fortus 250mc

The second 3D printer available at the ROBIN research group is the Stratasys Fortus 250mc. This printer creates parts with a technology which Stratasys calls Fused Deposition Modeling (FDM)[1]. A plastic called ABSplus is used in the creation of parts. The finished surface of models created with the Fortus 250mc are not as detailed as those created with the Connex 500. However, the plastic does not warp over time like the plastic from the Objet Connex 500 has sometimes been found to do.

During the work with building the CNC milling machine, the Fortus 250mc has been used exclusively. Unlike the Connex 500, the models from this machine have exhibited no tendencies to warp, and the models have consistently retained their intended shapes. The rougher surface finish created by the Fortus 250 does not present a problem with using the machine, as the surface finish is not critical to the operation of the CNC milling machine being built, and some surface roughness is acceptable.

## 3.4 Arduino

The Arduino is an open hardware microcontroller platform. It consists of a printed circuit board with a microcontroller and all the supporting hardware necessary to use the microcontroller mounted on it. All of the designs for the PCB (printed circuit board) layouts are freely available. A simple IDE (integrated development environment) has been created, and many code libraries have been written (*Arduino website* 2005). This has simplified working with microcontrollers, and has opened up the world of microcontrollers to people who would otherwise have been deterred by the complexity of working with them.

The Arduino is an excellent platform for simple microcontroller tasks, as it is affordable and easy to work with. The Arduino comes in many variations, and it is either based on an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. An Arduino will be used to control the CNC milling machine that will be built. As the Arduino will be used to run the G-code interpreter Grbl, it will need to be based on the 8-bit Atmel ATMEGA328 microcontroller. The latest version of Arduino with this microcontroller is the Arduino Uno. However, the Arduino chosen for the work with this thesis is the Arduino Duemilanove, simply because there are a lot of these already available at the research group, and they work fine with Grbl. Fig. 3.1 shows an Arduino Duemilanove.

[1]The term Fused Deposition Modeling is trademarked by Stratasys. The RepRap project calls this same process by the alternative name Fused Filament Fabrication (FFF).

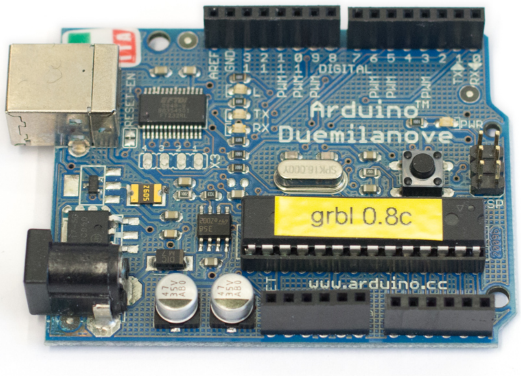



**Fig. 3.1** Arduino Duemilanove, programmed with Grbl 0.8

## 3.5 Grbl

Grbl is G-code interpreter written for the Arduino platform. It is written in the C programming language without going through the Arduino IDE to be able to squeeze as much functionality out of the limited memory of the Arduino as possible. Grbl takes its input from a computer connected through the Arduino's USB cable, translates the G-code into stepper motor control signals, and outputs these stepper motor control signals through dedicated output pins. The signals are then fed through stepper motor drivers to the stepper motors driving the CNC machine axes. Grbl is distributed as open source through the permissive free software MIT-license.

As the people behind Grbl themselves put it: "If the maker movement was an industry, Grbl would be the industry standard." (*Grbl Wiki* n.d.), referring to the fact that Grbl sits at the heart of many MakerBots and open source 3D printers. It is also the default way to control the Shapeoko CNC machine, which was presented in section 2.4.2.

When setting out with the master thesis project, the Grbl firmware was at version 0.8c. During the work with the thesis, this version was advanced to version 0.9g. The version 0.9 upgrade claims to increase stability and performance to production quality. In fact, upgrading from version 0.8c to version 0.9g did resolve some problems encountered during the work with building a CNC machine. The version upgrade also changes the communication serial baud rate up to 115 200 , up from 9600 in version 0.8.

## 3.6 Stepper motors

Stepper motors differ from regular electrical DC motors in that a full rotation of the motor is divided into a certain amount of steps. To turn the motor, a sequence of electrical signals need to be applied to the control wires of the stepper motor. As the electrical signals are applied, the shaft turns, one small step at a time, typically between 1 and 5 degrees. This allows the motor to

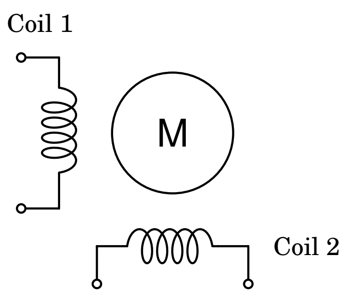



**Fig. 3.2** Wiring of a bipolar stepper motor

**Table 3.1** Step sequence of a bipolar stepper motor

| Seq. | 1a | 1b | 2a | 2b |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 0 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 |

be positioned precisely, making it useful in precision applications. There are two basic ways of wiring a stepper motor, as bipolar or unipolar. These two wiring schemes are detailed in the following sections.

### 3.6.1 Bipolar motors

Bipolar motors have two coils, both with a connection at each end. This gives a total of four wires to control the motor, as shown in Fig. 3.2. The sequence of input signals needed to advance the motor is shown in Table 3.1. When the motor has been through all four steps of the sequence, it starts over at the first sequence step. To turn the motor in the other direction the sequence order is reversed.

A bipolar stepper motor is more complicated to drive, but also gives better performance, than a unipolar stepper motor. The bipolar configuration is the one that will be used later when building the CNC milling machine described in chapter 5. The increased torque is beneficial when trying to avoid missing steps while the stepper motor is driving a milling tool through a potentially tough material. Although bipolar stepper motors require a somewhat more complex driving circuit, these circuits can be purchased as small preassembled driver boards without great expense.

### 3.6.2 Unipolar motors

The coils of a unipolar motor have center taps so that each coil has three connections, giving the stepper motor itself six wires. The two center taps may also be internally connected. In this case the stepper motor will have five

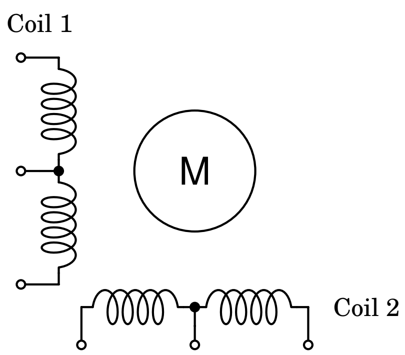



**Fig. 3.3** Wiring of a unipolar stepper motor

wires. If a stepper motor has eight wires, it may be wired either as a bipolar or a unipolar motor. A six-wire unipolar stepper motor is shown in Fig. 3.3. A unipolar motor may be driven by connecting each of the four driving signals through a transistor, making it less complex to drive than a bipolar motor.

The unipolar configuration will not be used in the CNC machine to be built. Bipolar stepper motors and their higher torque will be used instead.

## 3.7 Motor drivers

The signals for controlling the stepper motors typically come from a microcontroller or other source where the output current is limited. Motor drivers are placed between the microcontroller and the motor to provide the correct voltage level for the motor and to give the higher current needed.

### 3.7.1 H-bridge motor drivers

To drive a typical DC motor, an H-bridge motor driving circuit would be used. The H-bridge is named as such because of the way a typical circuit representation looks. In it, four switching transistors are arranged so that a positive or a negative voltage may be applied to a connected motor, depending on which of the switches are open and which are closed. Fig. 3.4 shows an H-bridge. Here, if signals are applied to the transistors $Q_1$ and $Q_4$ so that they are conducting, while transistors $Q_2$ and $Q_3$ are not conducting, a positive voltage will be applied across the terminals of the motor, and the motor will turn. If instead $Q_2$ and $Q_3$ are conducting and $Q_1$ and $Q_4$ are not conducting, a negative voltage will be applied across the motor, and the motor will turn in the other direction.

The motor is allowed to free-run if none of the transistors conduct current, and the motor will brake to a full stop if just the top two or just the bottom two transistors are conducting. To avoid a short-circuit, the two transistors $Q_1$ and $Q_2$ should not be conducting at the same time. Likewise $Q_3$ and $Q_4$ should not be conducting at the same time.

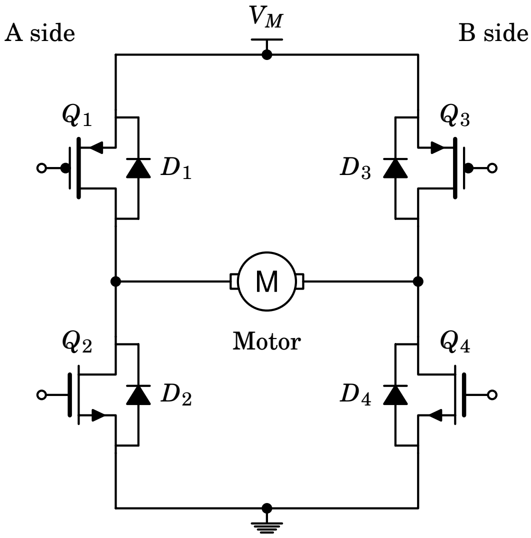



**Fig. 3.4** Simple H-bridge

The diodes present in the circuit are there to prevent damage to the transistors from inductive flyback voltages by providing a low-impedance path to ground for reverse voltages.

### 3.7.2 Stepper motor drivers

A stepper motor can not be driven simply by applying a positive or a negative voltage across it. To drive a bipolar stepper motor two H-bridge motor drivers will typically be used. Each of the H-bridges are connected to one of the coils of the stepper motor, as shown in Fig. 3.5.

In order to simplify driving the stepper motor H-bridges, a dedicated stepper motor driver may have some control logic that generates the control signals for the H-bridges. In this way the driver control signals are reduced to one signal for the desired direction of revolution, and another signal which advances the motor one step with every pulse. The task of generating the intermediary four-wire signals for the H-bridges is done by the control logic.

## 3.8 Computer-aided manufacturing

Computer-aided manufacturing (CAM) is the process in which a 3D model is translated into the machine toolpaths necessary to create that model. To create the toolpaths it is necessary to know what kind of machine will be doing the work and what type of machining tool bits are available. It is also necessary to know what kind of stock material will be used, as parameters for feed rates and spindle speeds must be specified. The toolpaths are generated in a series of operations, which may involve automatic changing of the tooling bit if the CNC machine is capable of this. The end result is typically output

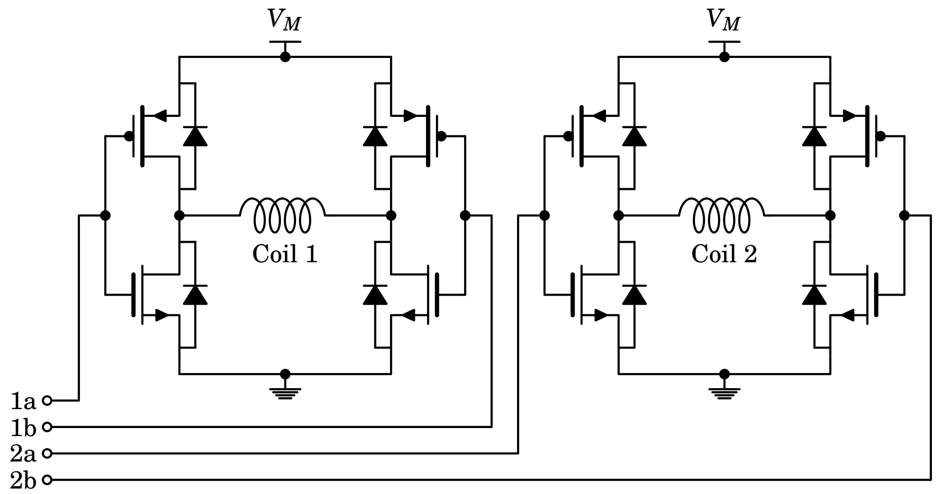



**Fig. 3.5** Two H-bridges connected to the coils of a stepper motor

as G-code, which can then be fed into a CNC machine, which does the rest of the work. Depending on the complexity of the CNC machine, the machine operator may have to change the milling bit or rotate the workpiece between operations.

HSM Xpress is the CAM software that has been used to create the toolpaths in the work with this thesis. HSM Xpress is the lighter version of HSM Works, which is free to use as long as it is used together with a licensed copy of SolidWorks. HSM Xpress, unlike HSM Works, is limited to 2.5D machining, meaning that machining may only be done along two axes at the same time, and as such is limited to machining in fixed Z-axis steps. Curved surfaces in the Z-direction are not available when limited to 2.5D machining. HSM Xpress is created as a plugin to be controlled directly from SolidWorks, allowing toolpaths to be created directly from a familiar environment. The ease of use that comes with the SolidWorks integration and the fact that the software is free for use with a licensed copy of SolidWorks is the reason why HSM Xpress has been used during the work with the thesis.

The fact that HSM Xpress can only generate 2.5D machining toolpaths will not be a problem, as many machine parts may be created in this way. It is a matter of designing the machine parts with this limitation in mind. Being limited to 2.5D machining gives an opportunity to be eased into the world of CAM by only having to simplest machining operations available. Where 3D machining is available, a whole array of more complex machining operations are available, and it may be a benefit to have acquired a good understanding of 2.5D machining before moving on to 3D machining.

### 3.8.1 Basic CAM operations

This is an overview of some common basic CAM operations. The described operations are all 2.5D machining operations which are available in the HSM

Xpress CAM software package. Although the term 2.5D machining refers to operations where the Z-axis is not moved at the same time as the X- and Y-axis, the helical motion of the bore operation technically moves all axes at the same time. These basic operations can be used to create many different machine parts. The different CAM operations are illustrated in Fig. 3.6.

**Face** A facing operation is often the first operation when manufacturing a part. This ensures that the top face of the stock material is perpendicular to the machining tool and takes away any eventual roughness in the finish of the stock material that the part will be created from. A large flat end milling tool or a special facing tool can be held in the spindle for this operation. A larger tool diameter reduces the number of passes necessary to face the surface and cuts down on manufacturing time.

**Drill** This operation is performed with a drill bit of the proper diameter inserted into the spindle. The operation can only create holes of the same size as the drill bit.

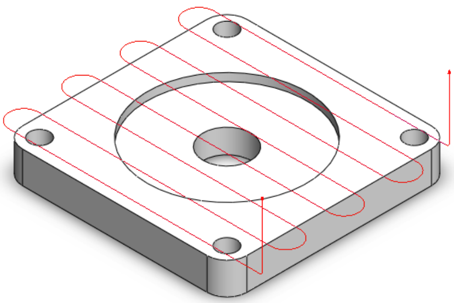
**Bore** This is an operation where a milling bit of one diameter is used to make a hole of a larger diameter in the workpiece. The spinning milling bit is lowered into the workpiece in a helical motion on a path which will create a hole of the desired size.

**Pocket** The pocketing operation is used to remove material from a large surface, often requiring the milling tool to take several passes to remove all of the material from the surface. The pocketing operation will typically leave material at the bottom of the workpiece without machining all the way through the part.

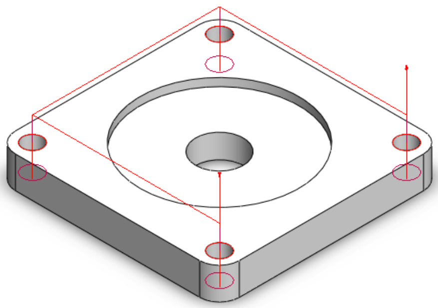
**Contour** The contour operation is used to follow the edge of a part. Both inside and outside contours can be followed. If the contour operation being performed is the outside contour of the part, the part is severed from the rest of the stock material. Depending on how the workpiece was fixed to the table, this operation may leave the piece free-floating without connection to the table. If this is the case, this operation is best reserved until the end of the manufacturing process to avoid refastening of the workpiece to the table. Another way to avoid having the part come loose from the stock material is to leave a number of tabs of material uncut along the contour to hold the workpiece in place.

### 3.8.2 G-code

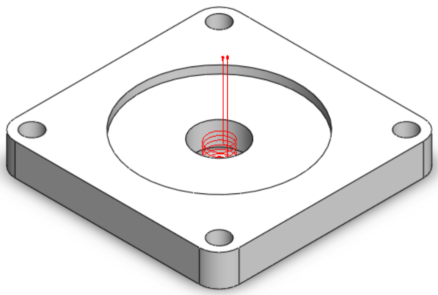
G-code is the most widespread numerical control language, and it is used with many different types of CNC machines. Commands starting with the letter G are preparatory commands, and this is where the term G-code originates from. Although there are other letters signifying other types of commands, G-code is often used as the name for the whole control language. Other letters are used for other types of commands. The letter M is used for miscellaneous function or action code such as spindle control. The letter T is used for tool



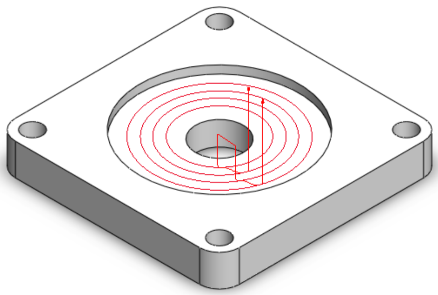
**(a)** Face.



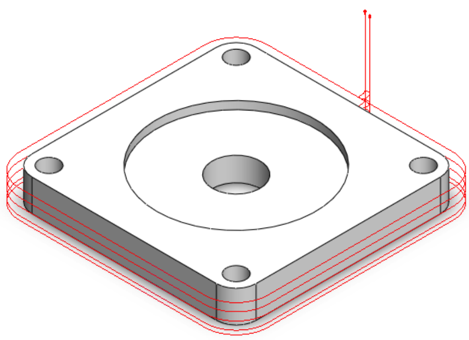
**(b)** Drill.



**(c)** Bore.



**(d)** Pocket.



**(e)** Contour.

**Fig. 3.6** Basic CAM operations.

selection. The letters X, Y and Z are used to give absolute or incremental position commands for the different axes. The position commands are given in conjunction with other G-codes. Individual axis coordinates may be omitted, and in this case the last coordinate given for that axis is used. The letters I, J and K are used to give arc centers in the X-, Y-, and Z-axes respectively. Some of the most used commands used in G-code programs are detailed below.

**G00** Rapid positioning. This command is used for rapidly moving the tool when not machining in the workpiece.

**G01** Linear interpolation. Movement when machining in the workpiece material. Position is given and the feed speed is specified.

**G02, G03** Circular interpolation, clockwise and counterclockwise. These two commands are used for moving the tool in an arc or a circle. The command is specified by either giving a start position, a specified radius or a center coordinate, and lastly the end coordinate.

**G17, G18, G19** XY, ZX or YZ plane selection. These commands are given when specifying which reference plane coordinates are given in reference to.

**G20, G21** Programming in inches or millimeters. These commands specify whether the program is given in inches or millimeters.

**M03, M04, M05** Spindle on, clockwise and counterclockwise, and spindle off. These commands turn the tool spindle on clockwise or counterclockwise, or off completely.

When using CAD and CAM software, the G-code is not manually programmed, but is generated by the CAM software after the human operator has specified the different machining operations at a higher abstraction level.

### 3.8.3 G-code example

An example of a small G-code program is given below. The program is written for a tool with a radius of 10 mm milling in a 5 mm thick material. First the tool follows a circular path made up of four quarter-circle arc movements, then the tool mills out a square path around the circle. The feed rate for milling is 150 mm/min, and 50 mm/min for plunging. When using a 10 mm milling bit the code will create a 40 mm square part with a 20 mm hole in the middle. The text enclosed by the parentheses are comments and are ignored by the G-code program. The whole program is enclosed by percentage characters to designate start the and end of the program. The resulting toolpath is shown in Fig. 3.7. In the toolpath illustration the starting point of the program is denoted by a small red downward pointing triangle, and the end point is denoted by a small green upward pointing triangle. Red lines mean rapid seek movements, black lines mean slower feed movements. A small coordinate system sign is placed at the origin.

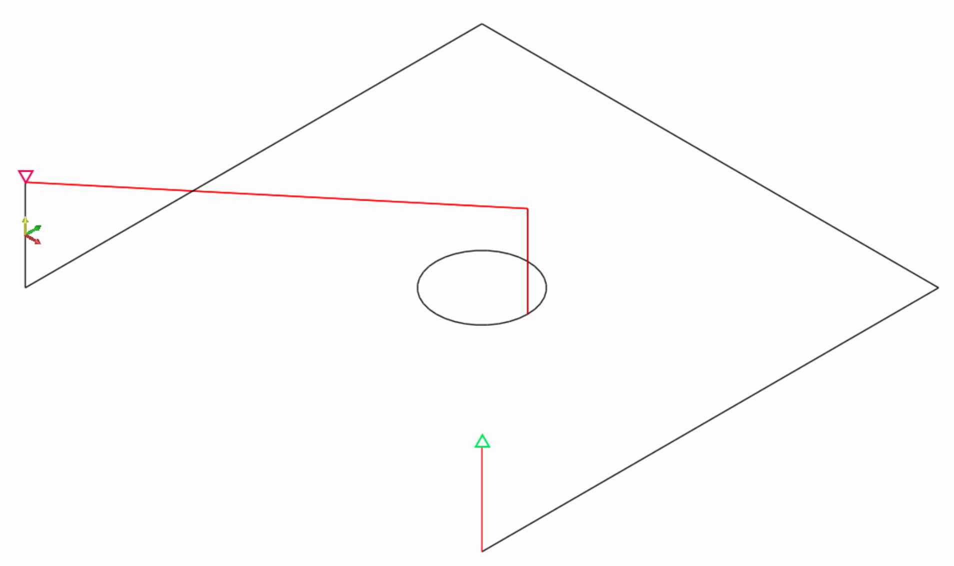


**Fig. 3.7** Toolpath for the G-code example

```
%
G90                        (Using absolute coordinates)
G17                        (Working on XY-plane)
G21                        (Using millimeters)

(Circle)
G00 X0 Y0 Z5               (Move to above the material)
M06                        (Spindle on clockwise)
G00 X30 Y25                (Move to starting point)
G01 Z-5 F50                (Plunge the tool down)
G03 X25 Y30 I25 J25 F150 (First quadrant of the circle)
G03 X20 Y25 I25 J25 F150 (Second quadrant of the circle)
G03 X25 Y20 I25 J25 F150 (Third quadrant of the circle)
G03 X30 Y25 I25 J25 F150 (Fourth quadrant of the circle)
G00 Z5                     (Retract the tool)

(Square)
G00 X0 Y0                  (Move to lower left corner)
G01 Z-5 F50                (Lower the tool into the material)
G01 X0 Y50 F150            (Move to top right corner)
G01 X50 Y50 F150           (Move to bottom right corner)
G01 X50 Y0 F150            (Move to bottom left corner)
G00 Z5                     (Retract the tool)
M06                        (Spindle stop)
%
```

# Chapter 4

# Experiments

In this chapter, experiments with two available educational CNC milling machines will be performed. This is done to gain some experience with running G-code on CNC machines, and to determine whether any of the two machines are able to improve themselves despite the fact that they are small and inexpensive machines.

At the time of writing, two small CNC milling machines are available at the ROBIN research group. The two milling machines are designed and built by Mats Høvin at the research group, and are intended to aid in learning to use CNC machines. The first of these is called the BabyMill. All custom components of the BabyMill are 3D printed in plastic. The BabyMill is shown in Fig. 4.1. The wires protruding from the back stem from an Arduino fixed to the machine. The Arduino mounted to the machine was not programmed with Grbl. To avoid disturbing whatever program may have been already programmed into the existing Arduino, the stepper motors of the machine were disconnected and reconnected to a second external Arduino using another set of stepper motor drivers. The second Arduino could be programmed with Grbl and the electronics could be reconnected to agree with the Grbl pinout. The electronics is described in section 4.1

The second machine is a further development of the BabyMill. It is called the MidiMill and is shown in Fig. 4.2. This machine makes use of milled aluminum parts in addition to 3D printed plastic parts. The aluminum parts are intended to increase rigidity and improve precision of the machine. The MidiMill has an increased working envelope compared to the BabyMill. Table 4.1 compares the two CNC machines.

## 4.1 Control Electronics

The two CNC mills are controlled by the same electronics circuit. They both use the same stepper motors, and can be powered by the same motor drivers. The EasyDriver stepper motor driver from SparkFun provides the power to drive the stepper motor and one driver is required for each stepper motor.

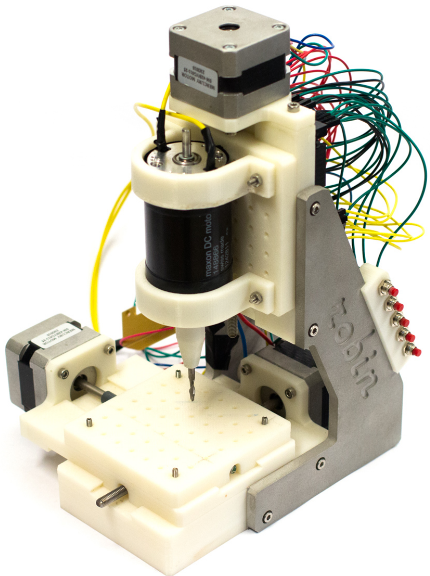


**Fig. 4.1** The BabyMill (designed and built by Mats Høvin)

**Table 4.1** BabyMill and MidiMill comparison

| | BabyMill | MidiMill |
|---|---|---|
| Stepper motors | NEMA 17 | NEMA 17 |
| Spindle | 12 V DC motor direct drive | 12 V DC motor driving Dremel spindle |
| Dimensions (mm) | 185 × 185 × 240 | 250 × 275 × 420 |
| Working area (mm) | 70 × 50 × 23 | 75 × 50 × 45 |

The EasyDriver can drive up to 750 mA per phase of a bipolar stepper motor (*SparkFun Electronics website* n.d.), which is enough to drive the NEMA 17 stepper motors used in the BabyMill and the MidiMill. The stepper motors have a current rating of 330 mA.

The EasyDriver has an input to toggle which way the stepper motor will turn, and another input which will advance the motor one step for every signal pulse it receives. When programmed with Grbl, an Arduino provides these same direction and step signals for each of the three axes for a CNC machine. The EasyDriver boards were connected to the Arduino via a prototyping breadboard.

### 4.1.1 Configuring Grbl

Grbl 0.8c was loaded onto the Arduino. Then it had to be configured so that Grbl would know the relation between steps of the motor and the distance advanced on the axis. The number calculated would be the same for all axes. The motors used have 1.8 degrees/step, which works out to 200 steps for a complete revolution of the motor. The motor drivers were not set up to use microstepping. The axis rods are standard ISO threaded M5 rods with a pitch of 0.8 mm per revolution. To calculate the number of steps per millimeter required by Grbl the following equation was used:

$$steps\_per\_mm = \frac{steps\_per\_rev \cdot microsteps}{mm\_per\_rev} \tag{4.1}$$

Inserting the relevant values into the above equation gives:

$$steps\_per\_mm = \frac{200 \cdot 1}{0.8} = 250 \tag{4.2}$$

Grbl also needs to know the default feed and seek rates for the CNC machine. The feed rate is the speed of the tool when being moved through a material, and the seek rate is the speed of the tool when repositioning from one spot to another. These speeds are normally determined in the G-code sent to the machine, but when running the machine manually the default rates will be used. The speeds were determined by trial and error by letting Grbl try to run the axes at different speeds, and setting the default seek rate at the highest speed at which the stepper motors could keep up. Setting the rates too high would lead to the motors stalling. The default feed rate was set slightly lower than the default seek rate, as this is a speed for moving through material. Table 4.2 shows the final configuration values used for Grbl, both the default values and the ones specifically modified for the BabyMill and MidiMill.

**Table 4.2** BabyMill/MidiMill Grbl 0.8c configuration

| Setting | Value | | Description |
|---|---|---|---|
| | Default | Modified | |
| \$0 | 250.000 | | X, step/mm |
| \$1 | 250.000 | | Y, step/mm |
| \$2 | 250.000 | | Z, step/mm |
| \$3 | 10 | | Step pulse, µs |
| \$4 | 250.000 | 100.000 | Default feed, mm/min |
| \$5 | 500.000 | 120.000 | Default seek, mm/min |
| \$6 | $0100\,0000_2$ | | Step port invert mask, binary |
| \$7 | 25 | | Step idle delay, ms |
| \$8 | 10.000 | | Acceleration, mm/s$^2$ |
| \$9 | 0.050 | | Junction deviation, mm |
| \$10 | 0.100 | | Arc, mm/segment |
| \$11 | 25 | | n-arc correction, integer |
| \$12 | 3 | | n-decimals, integer |
| \$13 | 0 | | Report inches, boolean |
| \$14 | 1 | | Auto start, boolean |
| \$15 | 0 | | Invert step enable, boolean |
| \$16 | 0 | | Hard limits, boolean |
| \$17 | 0 | | Homing cycle, boolean |
| \$18 | $0000\,0000_2$ | | Homing dir invert mask, binary |
| \$19 | 25.000 | | Homing feed, mm/min |
| \$20 | 250.000 | | Homing seek, mm/min |
| \$21 | 100 | | Homing debounce, ms |
| \$22 | 1.000 | | Homing pull-off, mm |

### 4.1.2 Explanation of Grbl 0.8 configuration values

This section explains the meaning of the different values given in the Grbl configuration setup. Note that as no limit switches are installed in the CNC machines, none of the homing or hard limit values apply, and they are left at their default values.

**Steps/mm** These values determine how many steps are needed to advance an axis one millimeter. There is a separate value for each of the three axes. The value is calculated as shown in equation (4.1).

**Step pulse** This is the duration of a single step pulse, given in microseconds. The pulses should be as short as the stepper drivers can handle.

**Default feed and seek** These values are the default speed at which the tool tip of the machine moves. Seek rate is the speed used when moving as fast as possible between two points (G-code G00). Feed rate is the speed for cutting motions (G-codes G01, G02, and G03). The default feed rate is usually superfluous, as the feed rate is usually specified in the G-code program. The speeds are given as millimeters per minute.

**Step port invert mask** Some CNC machines need the direction and step signals to be inverted, and this is the configuration value where the inversion of these values are specified. The value is given by modifying the individual bits of the 8-bit configuration byte. Step inversion for X, Y, and Z are given by bits 2, 3, and 4, respectively. Direction inversion for X, Y, and Z is given by bits 5, 6, and 7, respectively. Bits 0 and 1 are not used.

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| Dir. Z | Dir. Y | Dir. X | Step Z | Step Y | Step X | – | – |

**Step idle delay** Grbl disables the stepper motors when they have completed a motion and come to a stop. This value is the time Grbl will keep the steppers locked before disabling them, given in milliseconds.

**Acceleration** This value specifies the acceleration of the tool between the individual points of the toolpath. A low value gives a smoother acceleration, while a higher value allows the tool to reach the specified feed rate faster. The value is given in millimeters per second squared, mm/s$^2$.

**Junction deviation** Cornering junction deviation is a value which is used by the Grbl acceleration manager to determine how fast the tool can move through a path. Lower values give slower cornering while higher values give faster, less smooth cornering. The value is given in millimeters.

**Arc** Arcs are represented by a number of short lines. This value gives the length of each segment, specified in millimeters.

**n-arc correction** As each arc is made out of a number of short lines, and each of these lines introduce a small approximation error when the angle of the computed line is approximated, this value gives a maximum number of arc segments that can be performed before the cumulated approximation error needs to be computed so that it can be corrected. The value is given as an integer.

**n-decimals** Number of decimal places in floating point values Grbl reports.

**Report inches** Grbl can report the location of the tool at any time. This value specifies whether this value should be given in inches. This is a boolean value, and by default it is false, meaning Grbl will report in millimeters.

**Auto start** This is a boolean value which specifies if a G-code program should start immediately after upload, the default behavior, or if it should wait for the operator to hit a cycle start button.

**Invert step enable** Grbl has a stepper enable pin. By default a high value on this pin means enable, and a low value means disable. By toggling this boolean value, the behavior can be inverted.

**Hard limits** Upon enabling this boolean value, Grbl is notified of the presence of hard limit switches on the axes. If any movement causes any of these switches to be tripped, Grbl goes into alarm mode and shuts down all stepper motors, requiring an operator to reset Grbl.

**Homing cycle** If this boolean value is enabled, Grbl can use limit switches to perform a homing cycle on the CNC machine. This means that the machine operator can determine where the tool is accurately and repeatably every time the machine is used. Homing cycle can be used together with hard limits.

**Homing dir invert mask** Grbl assumes that the homing limit switches are at the positive ends of the axes. If the opposite is true, this value can invert the values. The entered value is a binary value, which works like the invert stepper mask described above.

**Homing feed and seek** The homing seek rate is the fast speed at which Grbl moves to find the limit switches during a homing cycle. After the limit switches have been triggered, the tool is moved to the machine zero point, and this is done with the slower homing feed speed. Both speeds are given in millimeters per minute.

**Homing debounce** This is a value to determine how long a homing limit switch is given to settle in to a switch value. This value is used to counter the mechanical noise which is inherent in all switches when switching from one value to another. The value is given in milliseconds.

**Homing pull-off** During a homing cycle, each axis is backed off a distance equal to this value when a limit switch is reached. This feature allows the homing cycle and the hard limits to share the same limit switches without an accidental hard limit being triggered during the homing cycle.

## 4.2 Experimenting with the BabyMill

With Grbl set up properly a program called UniversalGcodeSender running on a connected computer could be used to send signals to the Arduino, allowing the axes to be controlled manually. Controlling the individual axes manually provided a way to get a feel for how the machine responded, and to determine the working envelope of the machine. Some simple G-code files were then prepared with HSM Xpress to run the machine automatically.

One of the limitations detected when running the machine was that the working area was too small. It was decided that the milling machine should, if possible, try to mill out a printed circuit board to connect the Arduino to the stepper motor driver boards, to get rid of the temporary wiring of the breadboard. The circuit board was determined to be a suitably simple test to see if the machine could create a part for itself, but the working area was not big enough for the machine to be able to mill the whole board.

Another limitation was that while the machine was able to mill in foam, it would run into problems when milling a blank copper plated circuit board. The tool bit would flex due to the machine being made out of plastic. This prevented the tool bit from following a specified path with satisfactory precision. In addition, the spindle had not been equipped with a chuck, so the milling bit was limited to a 3 mm flat end mill bit, which would not be fine enough to mill the paths of the designed printed circuit board.

The BabyMill did not succeed in creating a part for itself, but the process of trying to make it work did provide experience in creating toolpaths with HSM Xpress and getting a machine to follow the created toolpaths.

As the BabyMill was able to mill in foam, this could have been exploited by first milling out a foam version of a desired part, and then making a mold from the foam part so that a copy of the part could be cast in a different material. This would necessarily lead to very small parts, and it was not tested out.

In light of the limitations uncovered when using the BabyMill, efforts of making the BabyMill create a part for itself were abandoned. Focus was shifted toward using the improved machine, the MidiMill, to create the same printed circuit board.

## 4.3 Experimenting with the MidiMill

From the initial tests with the MidiMill it was determined that this would be a more useful machine than the BabyMill. The spindle of the MidiMill is made out of the rotating part of a Dremel rotary tool, driven by a belt from a small but powerful DC motor. This ensured that the MidiMill had a chuck which would allow different milling bits to be used. Many of the structural components of the MidiMill have been milled out of aluminum flat

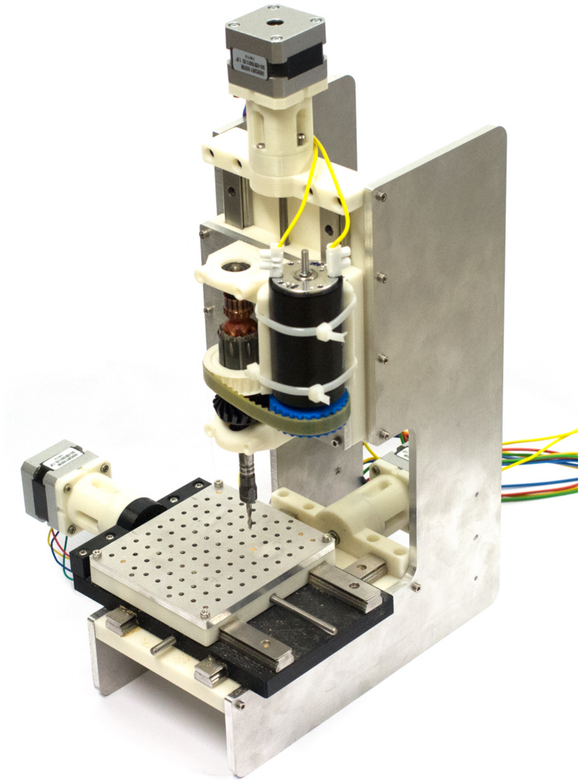


**Fig. 4.2** The MidiMill (designed and built by Mats Høvin)

stock, giving a more rigid machine than if everything had been 3D printed plastic.

The copper plated PCB blank intended for use with the machining test was a little too big for the machine work table, but some inventive mounting techniques were utilized to fix it in place. The PCB blank was laid on top of some sacrificial transparent acrylic plate to raise it up a little. The PCB blank could then be allowed to overhang the worktable slightly without colliding with the X-axis motor, increasing the reach of the X-axis. The milling setup is shown in Fig. 4.3, and in the right side of the image the overhang is shown where the copper workpiece being milled on extends beyond the worktable. The worktable is the aluminum plate with the equally spaced holes in it.

A very fine end mill bit was used for the actual machining of the PCB, but even then the spacing between the tracks turned out a little wide. An engraving bit with a pointed tip would have been the better choice of milling bit for this operation, as narrower paths could be created with that type of tool.

The resulting PCB is shown in Fig. 4.4. It is designed as a shield for the Arduino. The tracks are a bit uneven owing to the fact that there is some backlash present in the MidiMill machine. However, the populated PCB was able to perform its function, to connect the Arduino interpreting the G-code commands to the three drivers controlling the stepper motors, as shown in Fig. 4.5.

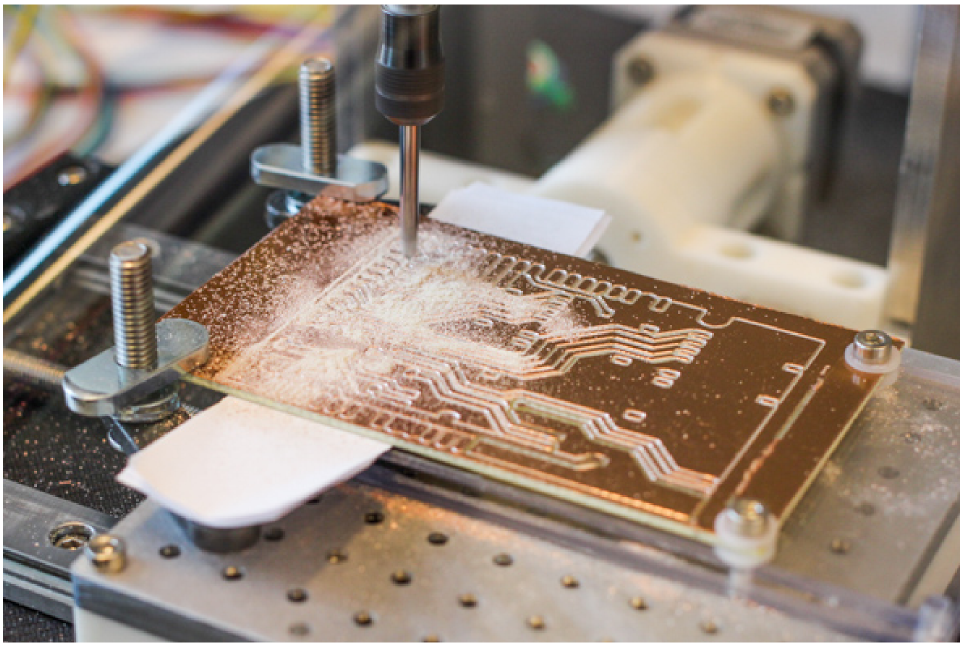


**Fig. 4.3** PCB milling in progress on the MidiMill.

The PCB created meant that a breadboard with temporary wiring could be substituted for more permanent, reliable and compact wiring to drive the MidiMill. This was the first experiment with a successful self-improvement.

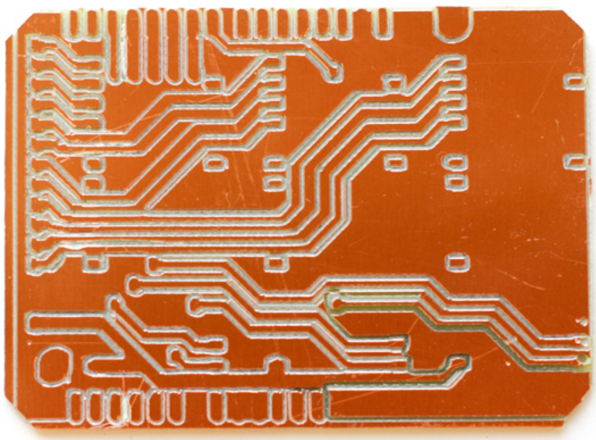


**Fig. 4.4** Test PCB created with the MidiMill.

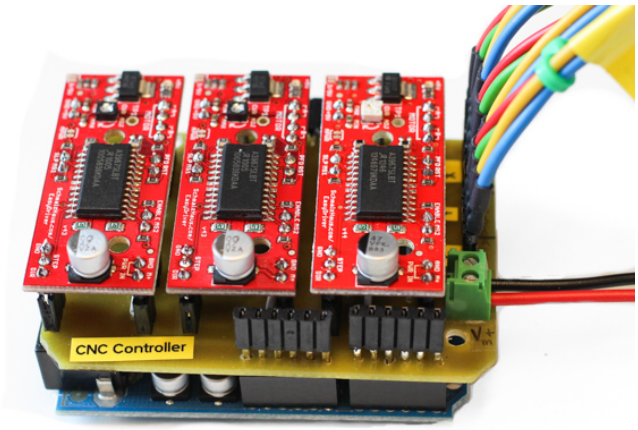



**Fig. 4.5** Populated test PCB in use between the Arduino and the stepper motor drivers.

# Chapter 5

# Implementation

This chapter is an answer to the first specific problem posed in the introduction of the thesis: "How can a self-improving CNC milling machine be created?" The process of building the CNC milling machine that will be used to manufacture its own parts is described. This is the main machine discussed in the thesis. Design inspiration is drawn from looking at how the BabyMill and the MidiMill are constructed, but the design is also to a large degree determined from what components were available at the ROBIN research group. The machine will be built by making use of components and materials that are already readily available at the research group, much like a hobbyist might make something out of parts he has lying around. The chapter starts out with describing the materials and components used to build the machine, and then goes into describing the individual parts of the machine.

## 5.1 Components and stock materials

The design of the CNC milling machine would be a balance of availability, cost and complexity. It was desirable for the machine to be made out of components that would be as easy as possible to get hold of. Stock materials such as aluminum profiles, solid woods, fiberboards and plastic sheets would all be good materials to use. They would all be readily available, either in stock, or just an order away. These are also not expensive materials.

The idea is to create the machine from a minimum of components and complexity, but also seeking to quickly create a machine which works. Once a working machine has been built, the machine can be improved, preferably letting the machine itself create the improved parts wherever possible.

Aluminum L-profiles were chosen as the building material for the mounting frame of the CNC machine. Steel L-profiles were considered as these could have been welded together to create a strong connection, but the research group had limited facilities for welding work, and this route was ruled out. Aluminum was chosen instead, and this had the added advantage that it turned out to be far less time consuming to cut to length.

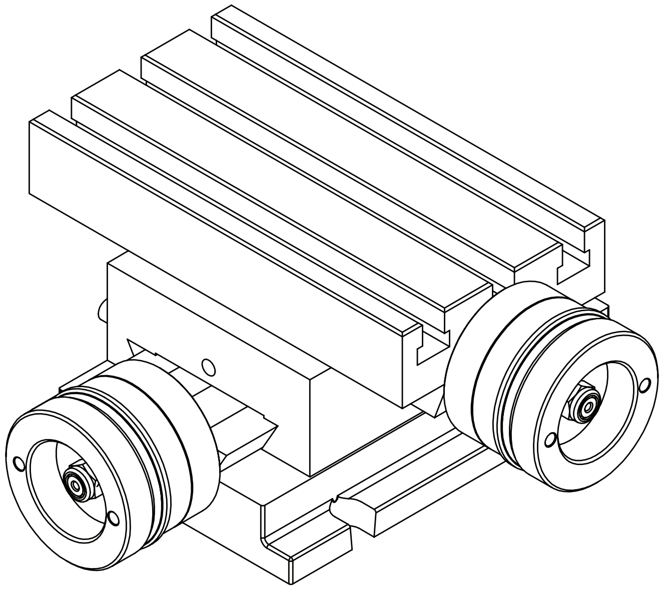


**Fig. 5.1** Coordinate table used for axes.

To create the movement of the individual axes, two coordinate tables similar to the ones shown in Fig. 5.1 were used. These consist of three main cast iron castings, linked together with two Acme set screws. Connected to the Acme set screws are turning wheels, allowing an operator to control the coordinates of the top surface.

## 5.2 Designing the CNC milling machine

Before any actual hands-on building took place, a 3D-model of the CNC machine was created. This was done in SolidWorks. First the initial given components were drawn individually as parts and sub-assemblies. Then the different components could be combined into a total model for the whole CNC machine. The first components to be modeled were the coordinate table and the stepper motors. These were components that were given at the start of the build, and they would form the basis of the finished machine and would dictate how the machine would be put together. The coordinate table is shown in Fig. 5.1 and the stepper motor is shown in Fig. 5.2.

### 5.2.1 Motor mount

The coordinate tables are fitted with manual turning wheels. These would be replaced by the stepper motors, so the coordinate tables were disassembled to determine how to connect the stepper motors to the coordinate tables. Two screws hold the manual turning wheel assembly to the bulk of the coordinate tables, so these screws could be used to hold a part which would in turn hold the stepper motor.

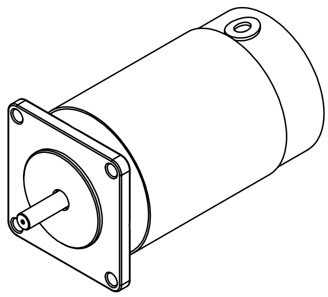


**Fig. 5.2** Stepper motor used in the machine.

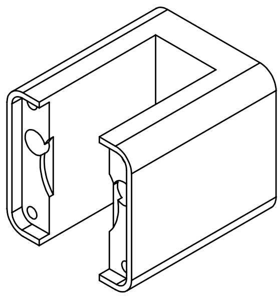


**Fig. 5.3** Custom stepper motor holder.

A custom stepper motor holder was designed. This would incorporate the bearings from the manual turning wheel and would fit onto the already existing screw holes. The stepper motor holder was designed to be 3D printed, and is shown in Fig. 5.3.

To connect the stepper motor shaft to the coordinate table axis shaft, some couplings were needed. The couplings used consist of two aluminum hubs connected by a spider made out of hard rubber. An exploded view of the coupling is shown in Fig. 5.4. One of the hubs was bored out to have an inner diameter of 6.35 mm, matching the shaft of the stepper motor. The other hub was bored out to 7.8 mm, matching the threaded end of the coordinate table shaft. The hubs would grip the shafts by means of a set screw.

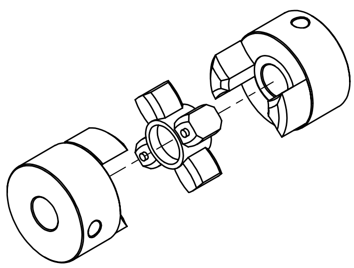


**Fig. 5.4** Shaft coupling consisting of two hubs connected by a spider, exploded view.

### 5.2.2 Frame

To mount the two coordinate tables in relation to each other, a frame was designed. The two planes of the frame would be mounted perpendicular to each other. This way, one coordinate table could be mounted on the horizontal plane, with the two axes representing the X- and the Y-axis. The other coordinate table would be mounted on the vertical plane where one of the axes would function as the Z-axis. The other axis of the second coordinate table would simply be bolted tight so that it would not move.

Before settling on using aluminum for the frame, it was designed to be built out of 50 mm × 50 mm L-profile steel bar, and the individual lengths were to be joined together by welding. When the time came to cut the different lengths of steel L-profiles specified in the generated cut list, it was discovered that available circular saw would only handle non-ferrous materials. To continue work with the frame of the CNC milling machine, it was decided that aluminum L-profile bars would be used instead of steel. As welding had been ruled out due to the limitations for this type of work at the research group, the frame would be joined together by nuts and bolts instead.

The pieces could have been cut with a hacksaw, but as a metal cutting chop saw was available this was used to save time. The ideal tool for making the cuts would have been a band saw, as using the thick circular blade of the chop saw takes longer to cut through material than a thin band saw blade would, but a band saw was not available at the research group. The holes needed to bolt the pieces together were drilled with a drill press, also available at the research group. This was quicker and more precise than using a handheld drill to make the holes, as it ensured that the holes were perpendicular to the surface. The pieces were then bolted together, using steel angle brackets to hold them to each other. Mounting holes were drilled in the frame, so the coordinate tables could be bolted on. Larger additional holes were drilled to accommodate the wiring to the stepper motors. The resulting frame is shown in Fig. 5.5.

### 5.2.3 Protective shield

To provide a measure of security when running the machine, clear polycarbonate sheets were used to construct a box which could be placed over the CNC milling machine during operation. This would protect against potential broken milling bits or material chips flung at high speed, and would also help contain the debris from the milling operation, keeping the work area tidy. In addition, the box may be able to dampen some of the noise generated by the milling when using the machine.

The clear polycarbonate sheets were cut by scoring the sheet several times with a retractable utility knife until the groove was deep enough so that the sheet could be broken by hand along the scoring mark. The sheets were held together by aluminum L-profiles to which the polycarbonate sheets were

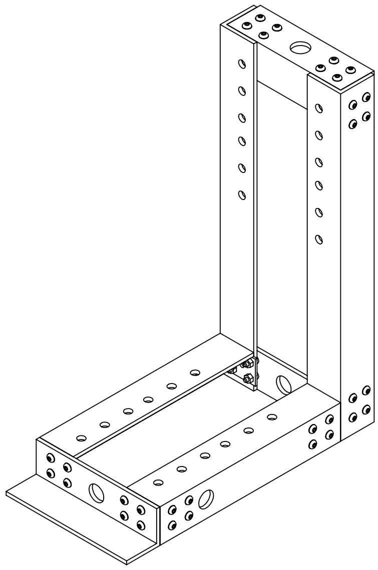


**Fig. 5.5** Frame of the CNC milling machine.

riveted. The resulting box has four sides and a top, but no bottom. The box is large enough so that during operation it can be placed over the CNC milling machine. The protective shield is shown in Fig. 5.6.

In addition to having a protective shield, safety goggles will be used during operation of the machine, as it is a cheap way to provide an extra layer of safety. Although the hope was to reduce the noise by using the protective shield, milling with the protective shield only reduces the noise minimally. Hearing protection will be used when milling.

## 5.3 Electronics

An Arduino programmed with Grbl would be used to control the CNC machine, in the same way that it was used with the BabyMill and the MidiMill. The thread pitch was larger on the coordinate tables compared to the lead screws of the smaller CNC mills. The coordinate table lead screws have a thread pitch of 2 mm/revolution, which when inserted into equation (4.1) gave the following steps per millimeter value:

$$steps\_per\_mm = \frac{200 \cdot 1}{2} = 100 \tag{5.1}$$

The thread per millimeter value is the same for all axes. The resulting configuration values for Grbl 0.8c are shown in Table 5.1. The meaning of the configuration values were discussed in section 4.1.2 and will not be repeated here. Note that any configuration values relating to homing or hard limit switches have been left at their default values, as the machine does not have any limit switches installed.

**Table 5.1** Grbl 0.8c configuration values for big CNC machine

| Setting | Value | | Description |
|---|---|---|---|
| | Default | Modified | |
| $0 | 250.000 | 100.000 | X, step/mm |
| $1 | 250.000 | 100.000 | Y, step/mm |
| $2 | 250.000 | 100.000 | Z, step/mm |
| $3 | 10 | 5 | Step pulse, µs |
| $4 | 250.000 | 120.000 | Default feed, mm/min |
| $5 | 500.000 | 150.000 | Default seek, mm/min |
| $6 | $0100\,0000_2$ | $1000\,0000_2$ | Step port invert mask, binary |
| $7 | 25 | | Step idle delay, ms |
| $8 | 10.000 | | Acceleration, mm/s$^2$ |
| $9 | 0.050 | | Junction deviation, mm |
| $10 | 0.100 | | Arc, mm/segment |
| $11 | 25 | | n-arc correction, integer |
| $12 | 3 | | n-decimals, integer |
| $13 | 0 | | Report inches, boolean |
| $14 | 1 | | Auto start, boolean |
| $15 | 0 | | Invert step enable, boolean |
| $16 | 0 | | Hard limits, boolean |
| $17 | 0 | | Homing cycle, boolean |
| $18 | $0000\,0000_2$ | | Homing dir invert mask, binary |
| $19 | 25.000 | | Homing feed, mm/min |
| $20 | 250.000 | | Homing seek, mm/min |
| $21 | 100 | | Homing debounce, ms |
| $22 | 1.000 | | Homing pull-off, mm |

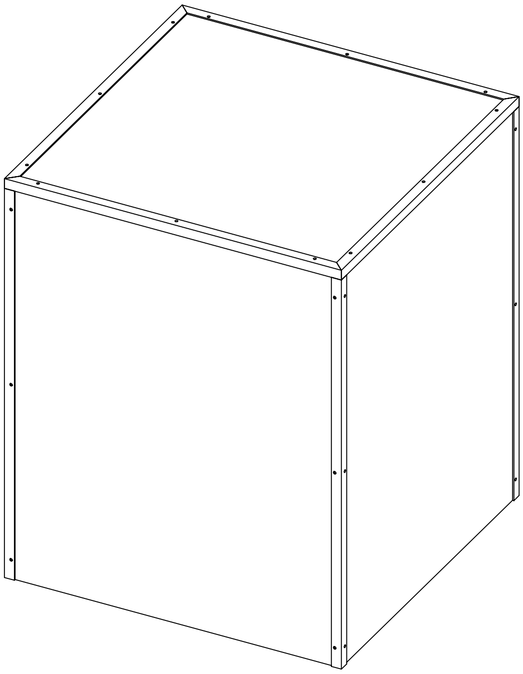


**Fig. 5.6** Protective box for the CNC milling machine.

As the stepper motors used with the big CNC machine would be bigger, they would demand more current than what could be delivered from the Easy-Driver stepper motor drivers used with the BabyMill and the MidiMill. The stepper motors used are NEMA 23 motors, with a voltage rating of 6 V and a current rating of 1.8 A. The motors have eight wires each. This means that there are eight internal coils, making it possible to wire the motors either in a unipolar or bipolar configuration. In this application pairs of internal coils will be wired together in series, making the motors behave as bipolar motors. This is done to take advantage of the extra torque provided when using stepper motors in this configuration. The wiring of the stepper motors is shown in Fig. 5.7.

As pairs of internal coils are wired in series in the stepper motors, each pair will be operated at a voltage level of 12 V.

### 5.3.1 H-bridge motor drivers

Some high power H-bridge motor drivers were available, but as these were made for driving DC-motors in general, they did not work by applying the step and direction signals. H-bridge workings were discussed in section 3.7.1. The motor driver carrier PCBs are made by Pololu Robotics and Electronics, and the motor driver chips on them are the *MC33887 5.0 A H-bridge with load current feedback* integrated chip from Freescale Semiconductor. The 5 A current rating refers to a chip which is sufficiently cooled by a heat sink (Freescale Semiconductor, 2012). Pololu specifies the current output as up to 5 A, but 2.5 A continuous, as the chips on their boards are not outfitted

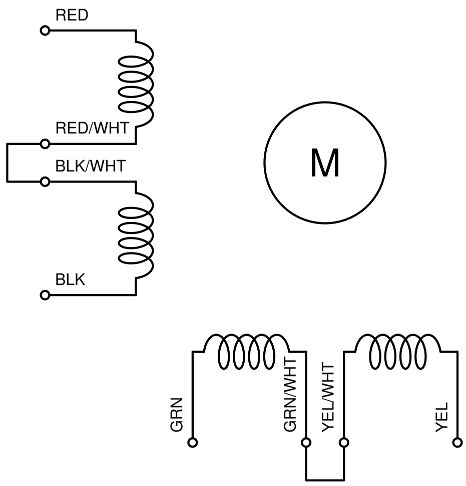



**Fig. 5.7** Wiring internal coils in series to create a bipolar motor.

with any external heat sink (*Pololu Robotics and Electronics website* n.d.). However, the continuous rating of 2.5 A is sufficient for the 1.8 A current draw of the stepper motors used, so it will not be necessary to mount an external heat sink to the chips on the motor driver boards.

### 5.3.2 Control logic

Two motor driver boards will be required for each stepper motor, as there is only one H-bridge circuit present on each board.

Additional control logic is needed to provide the necessary intermediary control signals between the step and direction outputs of Grbl, and the step sequence control signals of the H-bridge motor drivers. To generate the required signals a circuit was designed, utilizing two D-type flip-flops provided by a *7474* logic chip (Toshiba, 2007) and two exclusive-OR gates provided by a *7486* logic chip (Philips Semiconductors, 1990).

The step signal pulses are fed to the clock inputs of the flip-flops. The output of one of the flip-flops is XORed with the direction input signal, and the resulting signal is fed to the data input of the other flip-flop. The output of the second flip-flop is inverted before being XORed with the direction signal and then sent to the data input of the first flip-flop. The active-low set and reset inputs of the flip-flops are disabled by setting them high. A schematic representation of this is shown in Fig. 5.8. The output signals to be connected to the H-bridges are the inverted and non-inverted outputs from the two flip-flops.

This circuit was created so that the default Grbl software could be used. It is a modification of various similar circuits found online which use other components. Fig. 5.8 was created using electronics components available at the research group. The control signals could have been generated in software, but this would have meant modifying the Grbl software, and this was avoided

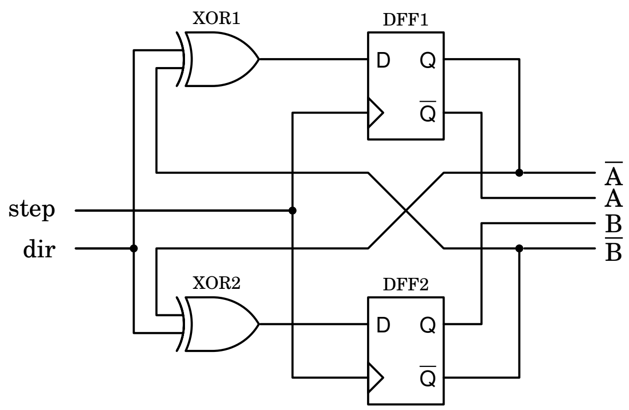



**Fig. 5.8** Circuit to generate motor H-bridge control signals from step and direction inputs.

as it seemed easier to create a small external electronics circuit instead of trying to modify Grbl.

If the Grbl software were to be modified to avoid the extra electronics hardware, the pin configuration of Grbl would have had to be altered. By default Grbl dedicates two ouput pins of the Arduino for each axis. This would have to be increased to four output pins for each axis if the signals were to be generated in software, and this would have been hard to make room for on the already crowded pin layout used by Grbl.

Alternatively, since the four needed control signals are just two pairs of inverted signals, software generation of the control signals could take advantage of this and just generate two control signals, one from each inverted pair. This would reduce the number of pins needed to output the signals from the Arduino from four to two, but would reintroduce the need for an external electronics circuit, as the remaining two control signals would have to be generated by a pair of external inverter components.

In conclusion, it seemed easier to create a small external electronics board than to try to modify the Grbl software, as Grbl is already highly optimized for the Arduino, and there is not too much space left for modifications.

## 5.4 Spindle

At this point all of the components of CNC milling machine have been mounted together. The two coordinate tables have been bolted to the aluminum frame at a 90° angle in relation to each other. The manual turning wheels on the coordinate tables have been replaced by the 3D printed motor mounts with the stepper motors attached. The shafts of the stepper motors have been coupled with the shafts of the coordinate tables, and a protective shield is in place to keep the operator safe.

The electronics of the machine have been wired up, and the Grbl-programmed

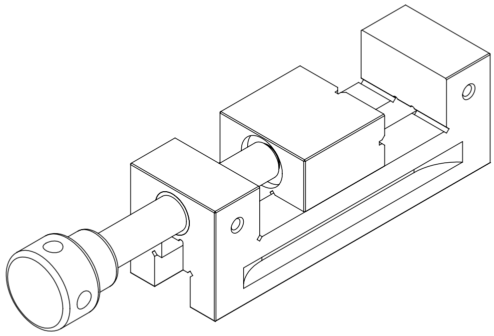


**Fig. 5.9** The small machine vise used with the CNC milling machine.

Arduino is able to move the axes of the machine. Initial motion testing of the machine in action was performed with a pen fixed to the Z-axis, drawing on some paper fixed to the table of the machine, but for the machine to be able to mill anything, a rotating spindle was needed.

A spindle was available at the research group. The spindle is a 230 V Kress 800 FME, and it is designated as a milling motor. A mounting block machined out of aluminum was also available at the research group. This mounting block slid right into the T-slots of the Z-axis coordinate table, and the spindle was fixed to the mounting block. The spindle provides 420 W of effective power. The range of the spindle speed is 10 000–30 000 rev/min, and this speed is selected by turning an adjustment wheel on the spindle.

The spindle is supplied with an 8 mm collet, accepting tools with an 8 mm shank, but this collet can be substituted for other collet sizes from 1 mm to 10 mm. Two additional collets have been purchased for the spindle. The size of these are 3 mm and 3.175 mm (1/8 inches). These collet sizes allow for using many of the milling bits available at the research group. The spindle will be turned on manually with the switch on the spindle itself when milling, and likewise be turned off when finished.

## 5.5 Workholding

The X-axis coordinate table is fixed to the frame at a height that allows for a machine vise to be fixed to the moving table. The machine vise allows workpieces to be clamped to the table quickly. The vise used is a small vise capable of opening its jaw 60 mm. It was already available at the research group, and a illustration of it is shown in Fig. 5.9. When used, it is clamped to the coordinate table using the T-slots in the table top.

As the Z-axis coordinate table is mounted at a height that allows for full range of motion when using a machine vise with the machine, the spindle does not reach all the way down to the workpiece if the workpiece is flat and the vise is not being used. Three different sets of coordinate table mounting holes have been drilled into the frame to allow for lowering the coordinate table when

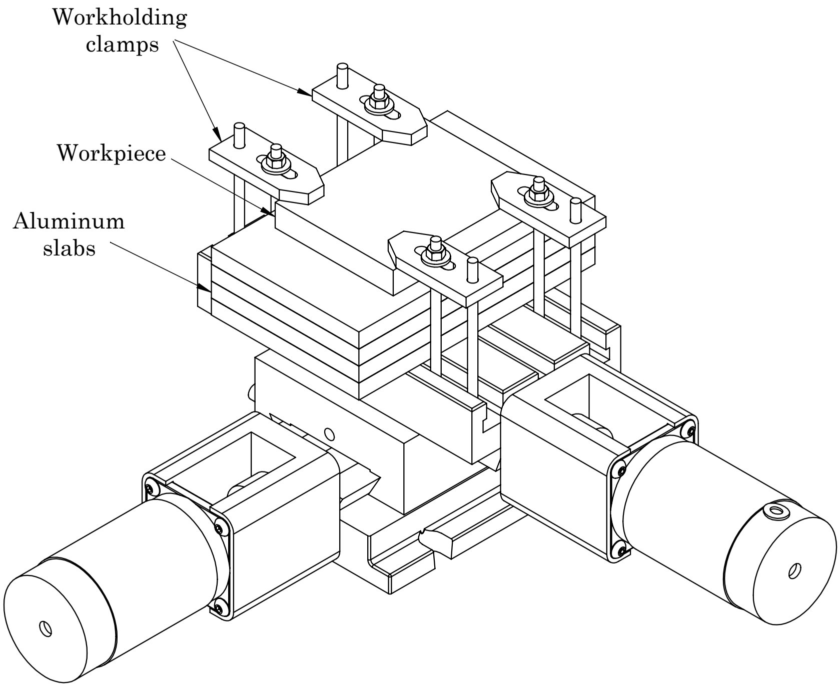



**Fig. 5.10** Workholding setup for flat low workpieces.

needed.

In practical everyday use, unfastening the Z-axis coordinate table, moving it down, and remounting it to the frame is a bit cumbersome. A workaround is to bring the workpiece up towards the spindle. This is done with the built machine by placing a number of 10 mm thick aluminum slabs under the workpiece. The slabs have approximate dimensions of 160 mm × 100 mm, and this is a good size for fitting to the coordinate table and still having room for work holding clamps to hold the slabs and a flat workpiece down. Fig. 5.10 shows this type of workholding setup in action. The workholding clamps are mounted to the table by threaded rods connected by T-nuts to the table T-slots. Nuts on the workholding clamps allow for tightening. The creation of a set of work holding clamps will be described in section 6.5.

## 5.6 Status of the CNC machine

A machine has been created, and some initial testing has been done with the machine. The next chapter will detail the steps taken to improve the machine after running the machine and determining that the precision must be increased for the machine to be useful.

Fig. 5.11 shows an exploded view of the CNC milling machine, giving an overview over how all of the components of the machine fit together.

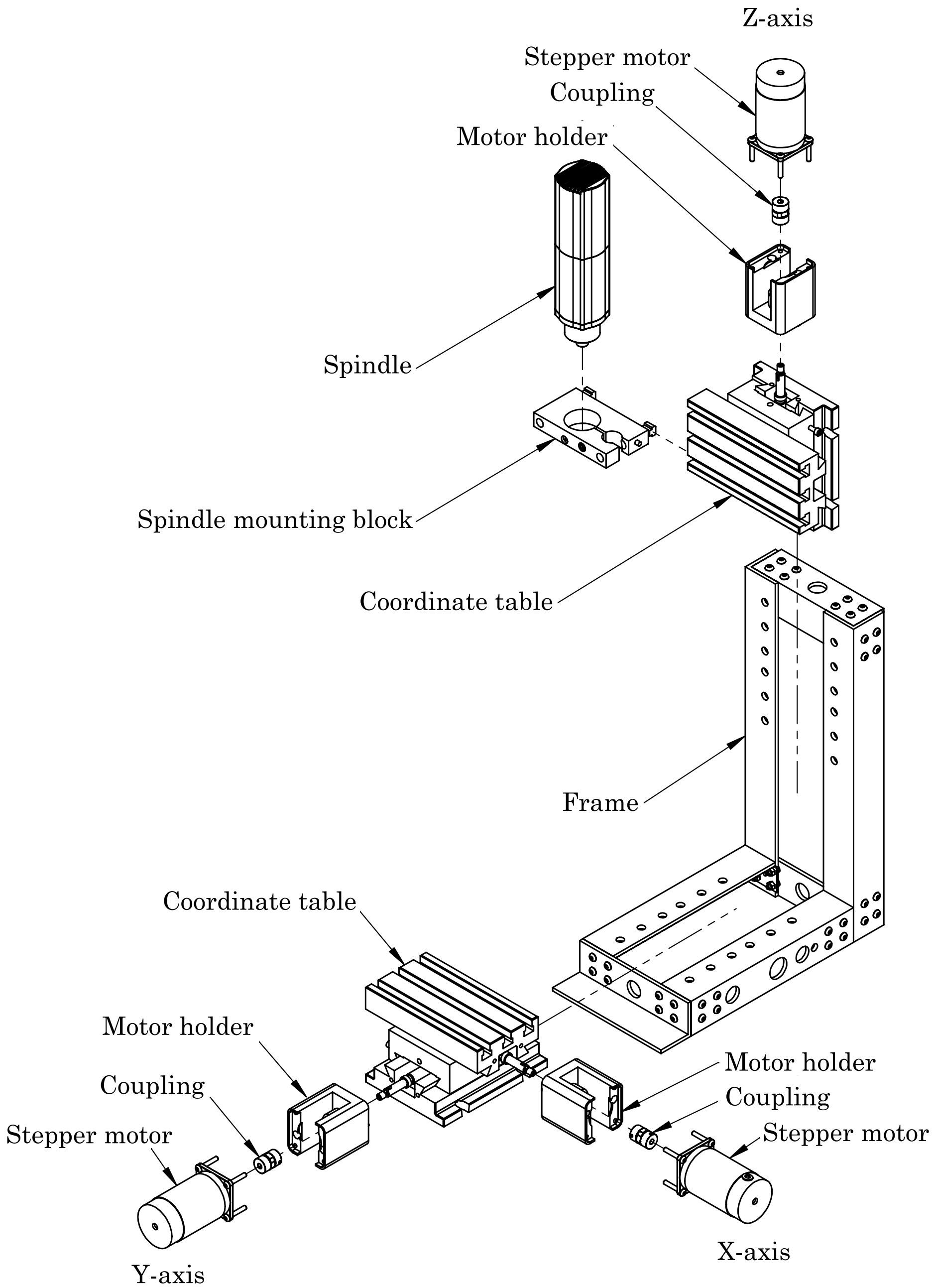



**Fig. 5.11** An exploded view of the CNC machine, showing how the different components of the machine fit together.

# Chapter 6

# Improving the Milling Machine

This chapter details the steps taken to improve the now built CNC milling machine. The improvements are performed after experimenting with how the machine performs at the different stages of improvement. One of the big issues with the machine is dealing with backlash, so this is described in detail in this chapter. After the machine has been rebuilt to a sufficient standard, it is used to experiment with milling in different materials, and some self-improvement is performed by creating components for the electronics assembly of the machine. A precision determination experiment is detailed in the chapter, and toward the end additional possible improvements are discussed. These last improvements are not performed on the machine, but are left as hypothetical improvements or perhaps future work.

The chapter will detail self-improvement performed by the machine, proving that the machine is in fact capable of self-improvement. The parts detailed, created for the machine by the machine itself, represent some of the parts most suitable for improvement when letting the machine create its own parts.

## 6.1 Stepper motor shaft coupling

As soon as the first version of the milling machine was built, and test milling began, some problems were discovered. The coupling hub which connected the coordinate table axis shaft to the shaft of the stepper motor would sometimes lose its grip. This is due to the fact that the coordinate table shaft has threads on the end, which stem from the mounting option of the manual turning wheel that the coordinate table is supplied with. The poor grip is pointed out in Fig. 6.1, which shows the cutaway view of the standard shaft coupling hub mounted onto a threaded rod. As the coupling mechanism tried to grip onto the threads it would sometimes slip, as there was not enough friction at the tips of the threads.

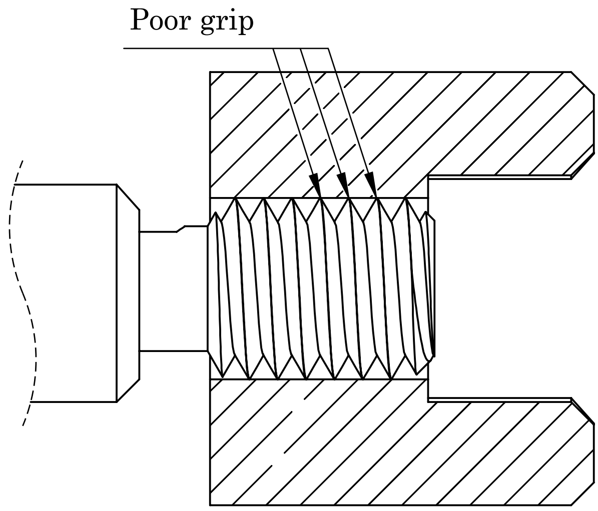
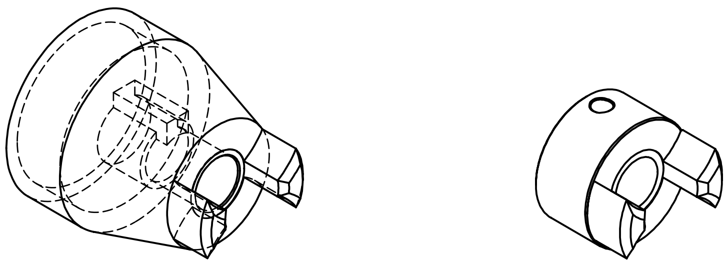



**Fig. 6.1** Threading leading to poor grip in coupling. Cutaway view.



**(a)** Custom shaft coupling hub. Hidden lines visible.

**(b)** Stock hub, shown for comparison

**Fig. 6.2** Shaft coupling hubs

To remedy the slipping grip on the axle, a custom component was designed and 3D printed. The custom component is a custom shaft coupling hub which utilizes the small key which comes default with the coordinate table. The off-the-shelf shaft coupling hub uses a set screw to tighten onto whatever shaft it is connected to. A sketch of the custom part is shown in Fig. 6.2a, where the hidden lines are shown to reveal the notch which receives the key. One end of the part slips into the hub coupling, and the other has a notch which accepts the tiny key inserted into the axle. This guarantees that the axle will not slip. For comparison, the standard hub is shown in Fig. 6.2b. Fig. 6.3 shows how the different components fit together in an exploded detail view of one of the shaft couplings.

## 6.2 Backlash

Lead screws are not machined to a precise tight fit. A very close tolerance in a lead screw would lead to substantial friction, rendering the lead screw

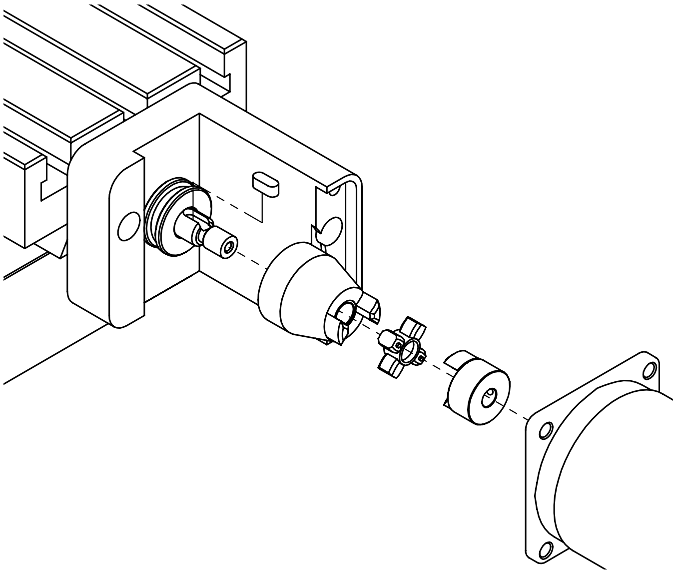


**Fig. 6.3** Exploded view of the shaft coupling, showing the placement of the custom hub.

unusable for a human operator. Instead, the screws are machined so that there will be some slack between the screw and whatever the screw is going into. In the building of the CNC machine, this posed a problem. As there was not a tight fit between the screw and the special nut connecting the lead screw to the coordinate table, there was some play in the axis direction, as illustrated in Fig. 6.4. This backlash present in the machine would result in imprecise operation.

Backlash is not a problem when using the coordinate tables manually, as a human operator is able to intuitively compensate for the lack of precision by just turning the rotational wheel further. However, when mounting stepper motors to the coordinate table in place of a turning wheel, the backlash becomes an issue to be addressed, as the machine will not be able to compensate without being explicitly told to do so.

### 6.2.1 Reducing backlash

Backlash can be compensated for in the software controlling the path generation of the milling machine and the movement of the stepper motors. Any time one of the stepper motors goes from rotating one way to rotating the other way, additional steps can be generated to take up the backlash before going ahead with the planned path. This will however slow down the system, as the paths will be more chopped up. For instance when creating a circular path, at the end of each quadrant arc, one axis motor would have to halt to wait for the other axis motor to take up the backlash before starting the movement again. If no backlash was present in the system, the first axis motor could just keep a constant speed while the second axis changed its direction.

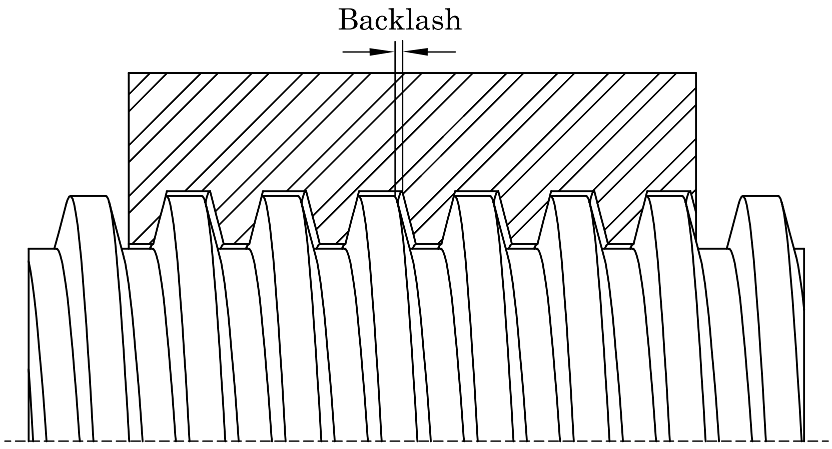



**Fig. 6.4** Backlash in a threaded rod

Instead of compensating for backlash on the fly in the step motor control software, this could be done by introducing another step in the software toolchain. After generating the G-code, the G-code could be run through a program which would modify the code based on a previously determined backlash. This would result in a modified version of the G-code in which the backlash compensation for that particular machine was embedded in it. One such program is NoLash, a standalone command-line tool for G-code backlash compensation. NoLash was originally written by Richard White for use with the RepRap, and has been ported to Java by Len Trigg (Trigg et al., 2013). As introducing another program in the software toolchain is a cumbersome way of compensating for backlash, this solution was not tested with the CNC milling machine.

Another way to compensate for backlash is to introduce a counterweight which would mean that the nut of the lead screw would always be drawn toward one edge of the lead screw threads. This happens naturally with the Z-axis stepper motor, as the weight of the coordinate table is pushed towards the lead screw threads by the force of gravity. As a result, backlash is effectively eliminated in the up/down direction. Implementing the same concept in the horizontal X- and Y-axes would be clunkier, as one would need to set up a system of ropes and pulleys connected in one end to a counterweight and in the other end to the coordinate table. This might become unnecessarily complex and space consuming.

Instead of using cumbersome counterweights to reduce backlash, the same thing could be achieved by spring-loading the nut through which the lead screw goes. A spring could be mounted between the lead screw nut and another accompanying nut that would sit on the same lead screw. By using the spring to press the two nuts away from each other, each nut would be pushed towards lead screw threads, causing the backlash to be taken up. For the X- and Y-axis neither of these backlash compensation techniques were used.

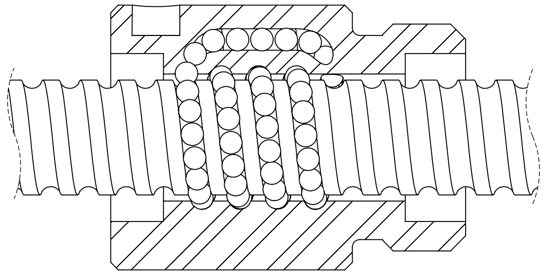


**Fig. 6.5** Cutaway view of a ball screw, showing the ball races on the ball screw and the recycling tunnel in the ball nut feeding the balls back into the races.

Instead ball screws were installed onto these axes.

A common way to reduce backlash in commercial CNC machines is to use precisely machined ball screws instead of traditional lead screws. Ball screws are screws that have ball races machined into the threads where a number of steel balls can roll, as shown in Fig. 6.5. The standard nut is replaced by a ball nut in which the steel balls are contained, and which also has a ball recycling system so that the steel balls are not simply ejected from the ball nut after running along the ball races. The steel balls lead to low friction in the ball screw, which again leads to high mechanical efficiency. Because of the tight machining tolerances of ball screws, backlash is reduced to the point of being effectively eliminated.

### 6.2.2 Implementing ball screws

To improve the CNC milling machine, the Acme lead screws would be replaced by ball screws in the X- and Y-axes. The ball screws chosen for the task are SKF ball screws with the product designation SD $8 \times 2.5$ R, meaning that the nominal diameter of the screw itself is 8 mm and the pitch of the screw is 2.5 mm.

The ball nut is threaded at one end for mounting it to a component. In the coordinate tables there was nothing to join to this thread, and the available space was limited, so a different mounting solution was devised. A custom nut holder was designed to be 3D printed in plastic. Made to resemble the original nut from the old Acme screw as much as possible, the custom ball nut holder consists of two parts that mount around the cylindrical ball nut and holds it in place. A stud in the custom nut holder fits into a hole in the ball nut, ensuring that the ball nut will not spin freely inside the part. The big cylinder of the nut holder acts as the mounting key to a hole in the coordinate table, and this cylinder is made to fit as snugly as possible into the hole, to prevent backlash. Fig. 6.6 shows an exploded view of the custom nut holder in relation to the ball screw.

Fig. 6.7 shows the completed ball screw assembly which replaces the original Acme screw. The ball screw has been cut to length and the ball nut has been threaded on, taking care not to release the balls inside. The nut holder has

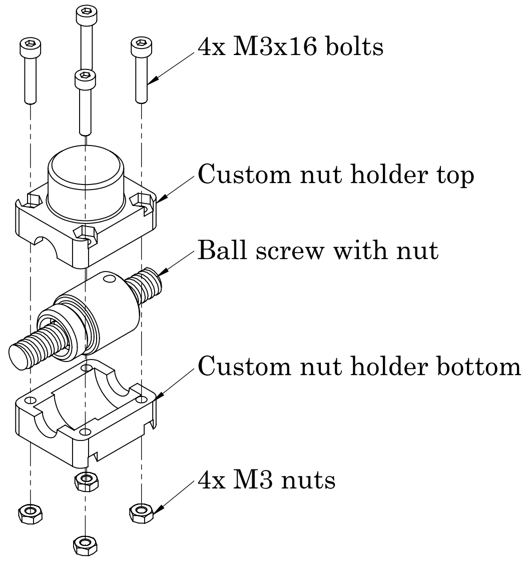



**Fig. 6.6** Exploded view of custom ball screw nut holder.

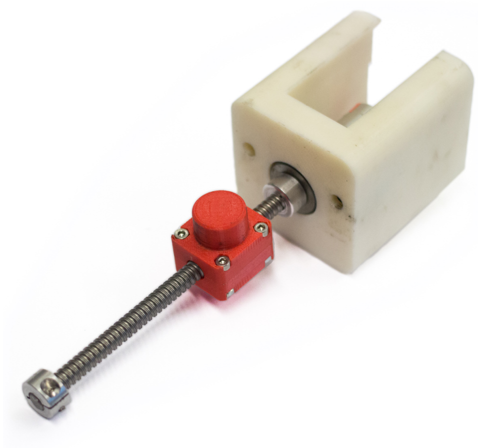


**Fig. 6.7** Ball screw axis assembly.

been mounted around the ball nut with four M3 bolts and nuts. An aluminum collar has been bored out to the correct bore to slip onto the ball screw, and it has then been installed at the end to prevent the ball nut from accidentally falling off. Aluminum collars have also been installed to hold the ball bearings at either side of the motor mount in place.

### 6.2.3 Configuring for ball screws

As the ball screw pitch of 2.5 mm/revolution is different from the pitch of the Acme lead screws, the value for steps per millimeter was recalculated. This value would be used for the X- and Y-axes where ball screws had been introduced. The Grbl firmware on the Arduino was updated to use the new value. The new value was calculated as follows:

$$steps\_per\_mm = \frac{steps\_per\_rev \cdot microsteps}{mm\_per\_rev} = \frac{200 \cdot 1}{2.5} = 80 \tag{6.1}$$

This meant that the configuration values \$0 and \$1 (X- and Y-axis steps per millimeter) in Table 5.1 would need to be changed from 100 to 80. The Z-axis value remains unchanged, as the ball screw upgrade was not needed there.

## 6.3 Coordinate table problems

After upgrading the X- and Y-axes to be driven by ball screws, further test milling was performed. The tests showed that the stepper motors would sometimes have a hard time driving the axes. This was most evident on the X-axis. As the stepper drove the X-axis towards its upper or lower limit, the table would run into problems with friction. There was enough friction for the stepper motor to skip steps when running, which would stop the workpiece and ruin the result.

There is a gib between the moving face of an axis and the face that it is moving relative to. A gib is a steel wedge which provides a bearing surface between two surfaces that move relative to each other. Three adjustments screws acting upon the gib may be tightened or loosened to provide more or less friction between the two faces. Placing the right amount of tension on the screws ensures that the axis only moves linearly along a single axis of movement. The faces are oiled to provide lubrication. One of the gibs of a coordinate table used in the CNC milling machine is shown along with its tightening screws in Fig. 6.8. In the figure, the top part of the coordinate table has been moved away. The gib is secured between the middle and top parts of the coordinate table. There are three notches in the gib that receive the tightening screws. The exploded view in the figure only shows one of the gibs of the coordinate table, but there are two gibs, one for each axis, and the three tightening screws for the other gib are also visible in the figure.

The problem which occurred and provided enough friction for the stepper motor to skip steps, stemmed from a difficulty of adjusting the gib screws to a good fit. If the screws were tightened to a snug fit with low enough friction when the coordinate table was centered, this would result in too much friction when the coordinate table was at either of its maximum extensions. If the adjusting screws were further loosened to avoid the skipped steps, the table would tilt slightly at either extension, due to the stepper motor acting as a weight on one side of the table. The tilt was most noticeable on the X-axis, as this axis sits at the top of the coordinate table. This tilt would have an influence on the precision of the machine, so the tilt would need to be eliminated.

It was discovered that the voltage provided to the stepper motors could be increased. The tests had been performed with 10 V provided to the motors to

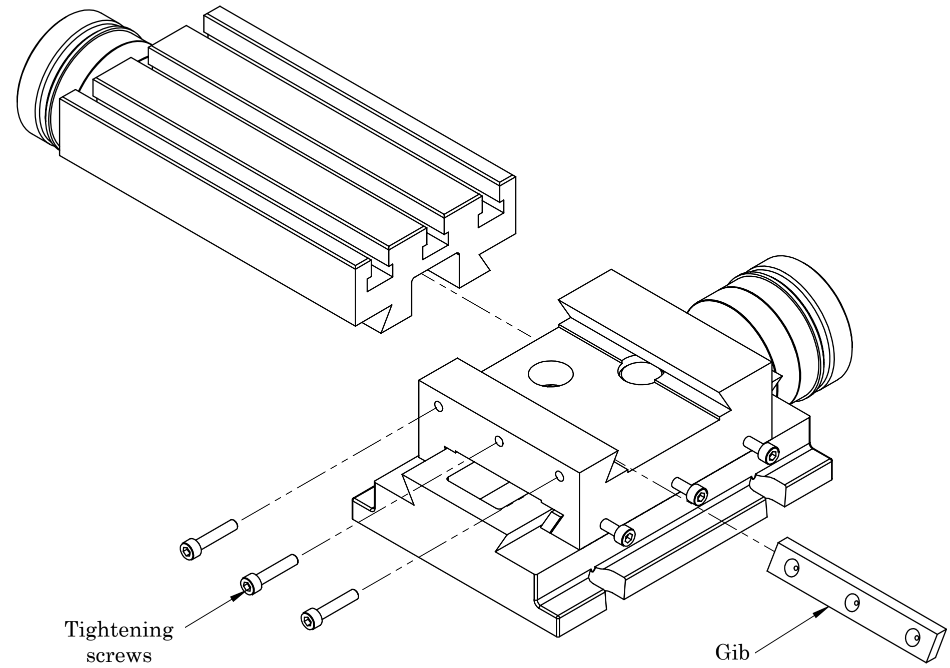



**Fig. 6.8** Exploded view of a coordinate table showing an adjusting gib and the accompanying tightening screws.

minimize the generated heat, and because this seemed to be a high enough voltage to drive the motors when running the machine through the air. However, when milling through some workpiece the motor voltage of 10 V was not enough. When the voltage was increased to 12 V, more current flowed through the coils of the stepper motors, providing more torque. This seemed to overcome the friction of the coordinate tables. The heat coming off the steppers was monitored, but it did not become a problem. The steppers are rated to handle this voltage level.

## 6.4 Grbl upgrade

With the voltage level of the stepper motors increased, the milling tests were continued. A new problem was discovered. When feeding the Grbl 0.8-equipped Arduino with a G-code program, the X-axis would sometimes refuse to run in the negative direction. Several unfruitful attempts were made to locate the source of the error, leading to a lot of frustration. Finally it was discovered that a new version of Grbl had been released during the work with the thesis, as mentioned in section 3.5. It was decided that the new version should be tested to see if it would resolve the problem.

The new version of Grbl, version 0.9g, was downloaded and installed into the Arduino. Grbl then had to be reconfigured, as the upgrade did not preserve the old configuration. In addition, the configuration values in Grbl had been changed for future-proofing purposes. The new configuration for Grbl version 0.9g is shown in Table 6.1.

Upgrading to the newer version of Grbl resolved the problem of the X-axis refusing to run in the negative direction.

### 6.4.1 Explanation of Grbl 0.9 configuration values

Many of the configuration values are the same in version 0.8 and 0.9 of Grbl, so the explanation of the different values given in section 4.1.2 still work for the newer version of Grbl, although some of the configuration setting numbers may have changed. Some differences in the configuration values are detailed below.

**Step port invert mask** Step and direction port masks are now separated into two separate values. This is a binary value where individual bits determine if the step signals should be inverted. Bits 0, 1, and 2 determine if the X-, Y-, and Z-axes are inverted, respectively.

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | Step Z | Step Y | Step X |

**Dir port invert mask** This is a binary value in which each of the individual bits determine if the direction of the axes should be inverted. Bits 0, 1, and 2 invert the X-, Y-, and Z-axis, respectively.

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | Dir Z | Dir Y | Dir X |

**Limit pins invert** This boolean value determines whether or not the limit input pins should be inverted.

**Probe pin invert** Grbl 0.9 introduces support for the G-code command G38.2. This is a command in which a probe is moved toward the workpiece and stops on contact. For this to work, one of the pins of the Arduino has been defined by the Grbl developers as the probe signal input pin. The probe pin invert configuration value is a boolean value that determines if the signal on the probe input pin should be inverted or not.

**Status report mask** This binary value determines which status information Grbl reports back to the operator when asked to do so. Machine position, work position, planner block buffer usage, and serial receive buffer status messages are enabled by setting bits 0, 1, 2 and 3, respectively. The running state is always returned with a status message, this cannot be turned off. By default, machine position and work position are reported together with the running state.

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| – | – | – | – | RX buffer | Planner buffer | Work position | Machine position |

**Table 6.1** Grbl 0.9g configuration values

| Setting | Value | | Description |
|---|---|---|---|
| | Default | Modified | |
| \$0 | 10 | 5 | Step pulse, µs |
| \$1 | 25 | | Step idle delay, ms |
| \$2 | $0000\,0000_2$ | | Step port invert mask, binary |
| \$3 | $0000\,0110_2$ | $0000\,0100_2$ | Dir. port invert mask, binary |
| \$4 | 0 | | Step enable invert, boolean |
| \$5 | 0 | | Limit pins invert, boolean |
| \$6 | 0 | | Probe pins invert, boolean |
| \$10 | $0000\,0011_2$ | | Status report mask, binary |
| \$11 | 0.020 | | Junction deviation, mm |
| \$12 | 0.002 | | Arc tolerance, mm |
| \$13 | 0 | | Report inches, boolean |
| \$14 | 1 | | Auto start, boolean |
| \$20 | 0 | | Soft limits, boolean |
| \$21 | 0 | | Hard limits, boolean |
| \$22 | 0 | | Homing cycle, boolean |
| \$23 | $0000\,0000_2$ | | Homing dir invert mask, binary |
| \$24 | 25.000 | | Homing feed, mm/min |
| \$25 | 250.000 | | Homing seek, mm/min |
| \$26 | 100 | | Homing debounce, ms |
| \$27 | 1.000 | | Homing pull-off, mm |
| \$100 | 250.000 | 80.000 | X, step/mm |
| \$101 | 250.000 | 80.000 | Y, step/mm |
| \$102 | 250.000 | 100.000 | Z, step/mm |
| \$110 | 500.000 | 150.000 | X max rate, mm/min |
| \$111 | 500.000 | 150.000 | Y max rate, mm/min |
| \$112 | 500.000 | 150.000 | Z max rate, mm/min |
| \$120 | 10.000 | | X accel, mm/s$^2$ |
| \$121 | 10.000 | | Y accel, mm/s$^2$ |
| \$122 | 10.000 | | Z accel, mm/s$^2$ |
| \$130 | 200.000 | | X max travel, mm |
| \$131 | 200.000 | | Y max travel, mm |
| \$132 | 200.000 | | Z max travel, mm |

**Fig. 6.9** Aluminum work holding clamps. First parts milled on the big CNC machine.

**Soft limits** Soft limits is a feature where Grbl will stop the execution of G-code if a given command will accidentally go beyond the available working area. The spindle and eventual coolant will stop, and an alarm will be passed. This boolean configuration value enables or disables the soft limits feature. The feature requires homing cycle to be enabled.

**Max rate** The maximum rate replaces the previous configuration values feed rate and seek rate. Maximum rate is specified for each individual axis, and travel speed of the CNC is limited such that none of the axes travel faster than its maximum rate. The speed values are specified in millimeters per minute.

**Max travel** These values, specified separately for each axis, specify the length of travel for each axis. The values are only used by the soft limit feature to determine if a motion takes the machine beyond its specified borders. The value is given in millimeters.

## 6.5 Milling in aluminum

With the new version of Grbl installed and configured, the CNC machine was finally able to do its first successful run of a simple G-code program. The G-code program describes milling workholding clamp out of a slab of aluminum, as this was considered a suitably simple part to create with the machine. The milled clamps, the first parts created on the machine, are shown in Fig. 6.9. They were milled with a feed rate of 150 mm/min, stepping down the toolpaths in the Z-axis direction with 1 mm for every pass. The resulting work holding clamps can be considered a self-improvement by the machine, as they are an accessory that will frequently be used for work holding during future milling with the machine.

Further examples of milled objects are displayed in the appendix, chapter C.

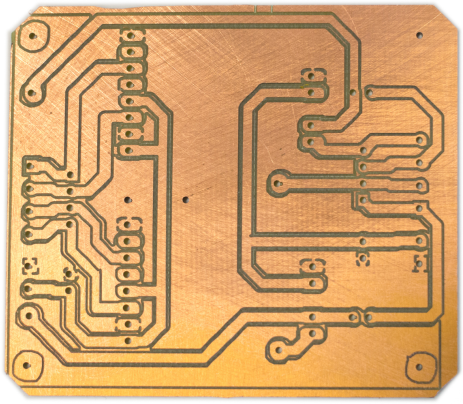


**Fig. 6.10** First PCB milled on the big CNC machine.

## 6.6 Milling simple printed circuit boards

After successful milling in aluminum, the CNC milling machine was tested with a part requiring higher precision. A small printed circuit board was designed with the free version of Eagle PCB. The board was designed to be as simple as possible, being a single-sided board with wide tracks. This was to increase the chances for the milling machine to be able to create a usable result.

The only components on the board are some custom designed sockets for H-bridge motor drivers, and a number of connectors. There is room for two motor drivers on the board, enough to drive a single stepper motor. The function of the PCB is to replace the temporary wiring of using a breadboard to connect the electronics.

The G-code for milling the PCB was generated by a plugin for Eagle called PCB-GCode. The board was milled with a 0.2 mm 45° engraving V-bit, meaning the mill tapers to a point. The milling depth was set to 0.2 mm. This was enough to mill through the 35 µm copper of the PCB even though the milling surface was slightly tilted. The PCB was made in three operations with three different milling bits. First the paths were milled, then the holes were drilled, and finally the board outline was milled out.

The resulting PCB is shown in Fig. 6.10. Some inaccuracy is evident due to X-axis backlash, but the finished board works and will be used with the CNC machine. As the board houses two H-bridge drivers, enough for a single stepper motor, two more PCBs like this were be created for the machine. They were left on the same physical board, and will provide the motor driver sockets for the remaining two stepper motors. As these boards get rid of some of the temporary electronics wiring for the CNC machine, they represent a self-improvement performed by the machine.

## 6.7 Increasing portability of the machine

As milling PCBs turned out well, another printed circuit board was designed. This PCB would be double-sided and would house the integrated circuits necessary to implement the control logic circuit described in section 5.3.2. This control logic PCB in combination with the three driver socket PCBs eliminates the need for a breadboard to house temporary electronics, and the electronics can be mounted onto the CNC milling machine itself, making the milling machine a self-contained unit which will be easier to move around.

Fig. 6.11 shows the temporary electronics wiring mounted to a breadboard, and Fig. 6.12 shows the improved permanent electronics assembly. The electronics assembly shows the Grbl-programmed Arduino in the top left of the image. The custom created control logic board is shown in the top right of the of the image. This PCB for the control logic board represents a self-improvement by the CNC milling machine, as it was milled on the machine itself. The bottom part of the figure shows the motor driver socket boards, each populated with two H-bridge motor drivers. There are three motor driver socket boards, though two of them have not been separated from each other. These single sided boards are as previously mentioned in section 6.6 another self-improvement by the CNC milling machine, as they too have been created on the machine itself. Each of the three motor driver socket boards houses two H-bridge motor driver boards. These are the smaller green rectangular PCBs.

The stepper motor wires feed into the terminal blocks at the bottom of the image in Fig. 6.11. This assembly houses all the electronics needed to drive the CNC milling machine, and has been mounted to an aluminum plate to simplify mounting the electronics to the back of the machine frame. All of the terminal blocks have been labeled with signal names and colors so that it is easy to understand how the individual boards connect together. The whole electronics assembly was mounted to the aluminum plate. The plate was was cut out on a circular saw, and the mounting holes were drilled with a handheld drill.

Seeking to further increase the portability of the machine, the clunky lab power supply that had been used in the initial test runs of the milling machine was also replaced. This lab supply is shown in Fig. 6.13. The lab supply is able to provide a DC voltage of 1–15 V, 60 A, which is overkill for this application (there is a text specifying maximum 5 A on the front of the enclosure, but there is another power outlet at the back for 60 A output). Instead of the lab supply, a discarded older computer power supply was used for power. The computer power supply is able to provide 12 V, 12.5 A, which is still plenty of power to provide the necessary 12 V, 5.6 A needed by the stepper motors. The computer power supply had been previously modified with a new end connector. The computer power supply is shown in Fig. 6.14.

A mating power socket matching the computer power supply was fixed to a hole in the machine frame with the help of a custom made two-part power

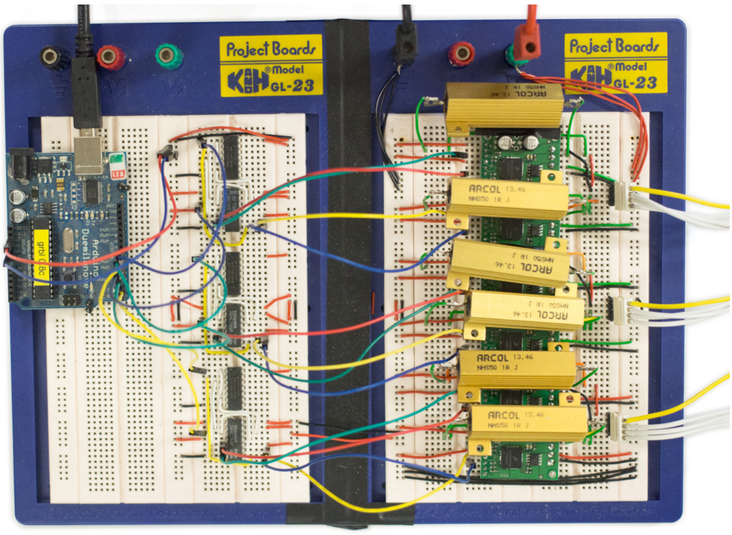



**Fig. 6.11** Temporary breadboard wiring.

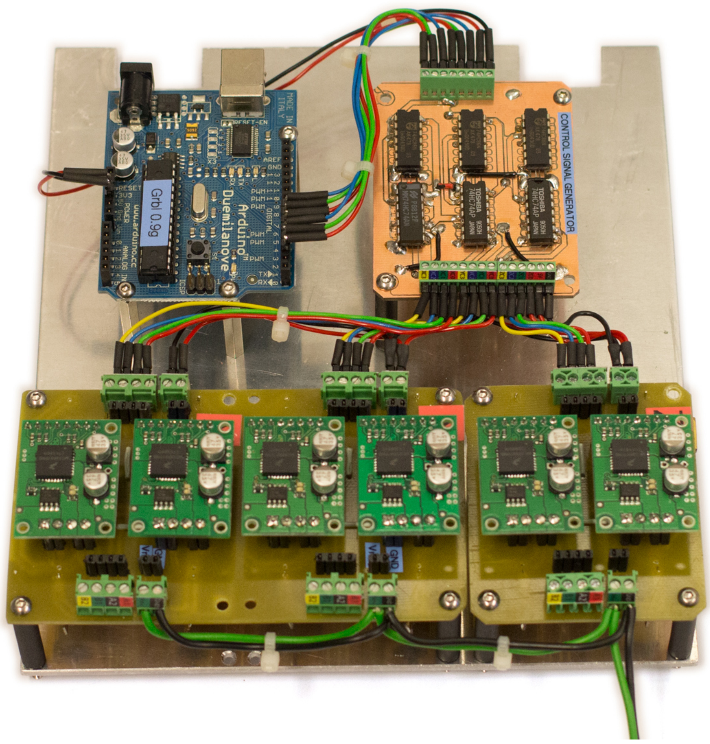



**Fig. 6.12** Improved permanent electronics assembly.


**Fig. 6.13** Lab power supply used during initial testing of the machine.

**Fig. 6.14** Modified computer power supply intended as the permanent power solution for the machine.

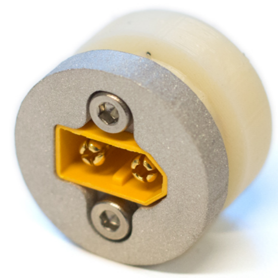


**Fig. 6.15** Assembled power connector holder. Rear part has been 3D printed.

connector. The power connector holder was made out of a 3D printed part and an aluminum part milled on the CNC milling machine, representing another small self-improvement by the machine. The assembled power connector holder is shown in Fig. 6.15. The power connector on the machine eases connecting and disconnecting power, and increases the portability of the machine. A custom USB connector holder was also created by 3D printing, and this connector was placed in a hole beside the power connector holder to add easy access to the USB connector. The USB connector is an extension of the Arduino USB connector used by Grbl to receive G-code from a connected computer.

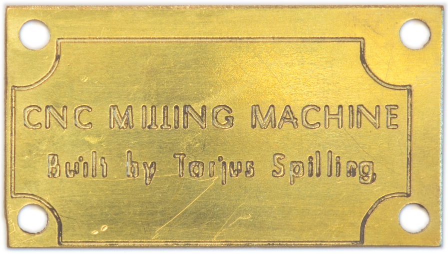
## 6.8 Determining the precision of milling

To test the precision of the milling machine, a brass nameplate was designed. It is meant to be fixed to the CNC machine, indicating who has built the machine.

The plaque was milled out of 1 mm thick brass plate. The plaque dimensions are 45 mm × 25 mm, allowing it to be mounted to the 50 mm width of the aluminum frame elements. The height of the letters on the plaque is 2 mm, demanding high precision of the milling machine. The engraving was done with the same 0.2 mm 45° engraving V-bit as used when milling the PCB.

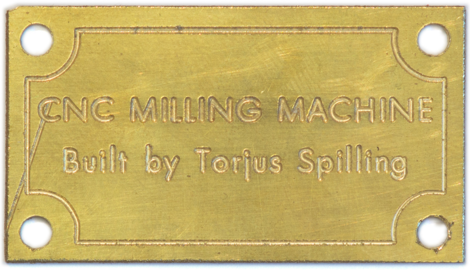
The resulting plaque is small enough that it shows significant distortion of the letters. This is shown in Fig. 6.16a. Notice especially the misalignement of the lower bar of the character "L" in the word "MILLING", or the misalignment of the dot over the character "i" in the word "Built", magnified in Fig. 6.16b. The nature of the distortion is such that backlash of the X-axis is the likely cause. To the naked eye the error of the X-axis is clearly visible, while the Y-axis does not show any error discernible with the naked eye. Looking through a loupe with built-in measurement lines, the error in the X-axis direction was determined to be 0.30 mm.

To minimize the documented X-axis error, the 3D printed ball screw nut




**(a)** Bad plaque.

**(b)** Bad plaque magnified.




**(c)** Good plaque.


**(d)** Good plaque magnified.

**Fig. 6.16** Two versions of the milled brass name plaque. The top, bad, plaque was milled before adjusting X-axis error. The bottom, good, plaque has been milled after the adjustment.

holder will be substituted with another similar part with a tighter fit. This procedure was previously performed on the Y-axis, but was not done with the X-axis, as it was believed that the X-axis was precise enough. The ball screw nut holder was documented in section 6.2.2.

After the milling machine was modified to improve the X-axis precision, a new plaque was milled. The new plaque is depicted in Fig. 6.16c. Looking through a loupe again, the X-axis error was now determined to be 0.08 mm, an improvement of 0.22 mm over the previous result.

## 6.9 Further possible improvements

To further perform some self-improvement, the CNC milling machine could now have been put to work to create aluminum versions of the 3D printed parts currently in use on the machine. However, the machine functions well even without replacing these parts, although the lifespan of the parts may be lower than if they had been machined out of aluminum. Some proposed aluminum alternative designs are presented below, the parts having been redesigned to be as easy as possible to manufacture on the CNC milling machine. These parts have not been made on the milling machine.

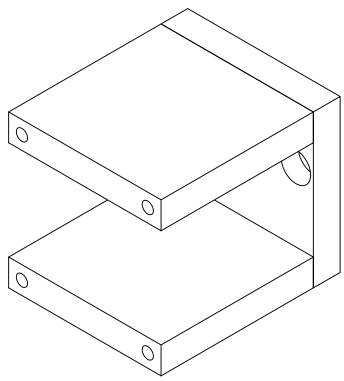


**Fig. 6.17** Proposed design for an all aluminum motor mount.

### 6.9.1 Stepper motor mount

If new versions of the motor mounts were to be made out of aluminum, one could imagine a design such as the one shown in Fig. 6.17. It has been simplified from the design of the 3D printed version to cut down on machining time. It is made out of 10 mm aluminum flat stock machined into three separate blocks and bolted together with four screws inserted from the back.

A further design change from the 3D printed version is that the solid sides now are located at the top and bottom instead of at the two sides. This is to shield the shaft coupling located inside from chips generated during milling. The four visible holes at the front are tapped, and are used for mounting the stepper motor.

### 6.9.2 Custom Z-axis shaft coupling

The custom shaft coupling discussed in section 6.1 was created in 3D printed plastic. This custom coupling is only still used in the Z-axis coupling, as the X- and Y-axes were converted to use ball screws which instead use a stock coupling hub with a set screw, in which the center hole has been bored out to dimension.

After the plastic custom shaft coupling was made, another potential mounting option was envisioned. If the hole of the stock coupling hub (shown in Fig. 6.2b) were to be bored out so that it had a wider inner diameter, then the resulting hole could be tapped with a thread matching the treading of the coordinate table shaft. Combined with either the set screw supplied with the coupling hub, or a locking nut tightened up against the coupling hub, this would presumably eliminate the slippage discussed in section 6.1. This design could be performed on a purchased stock coupling hub, or the whole coupling hub could be machined on the built CNC milling machine from scratch from aluminum round stock.

This proposed alternative design would probably be a superior design to the

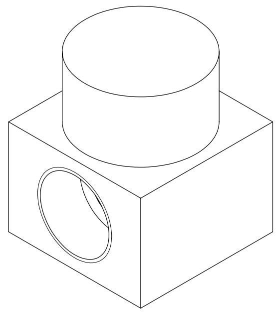


**Fig. 6.18** Proposed design for an aluminum ball screw mount.

plastic one already created. It remained a hypothetical solution, as the already created plastic custom coupling hub was working adequately for the time being, and attention had already been turned to other aspects of the machine.

### 6.9.3 Ball nut housing

The ball nuts used with the ball screws on the X- and Z-axes are threaded on one end. The thread is intended for use when fastening the ball nut to whatever it will be acting upon. In the 3D printed plastic ball nut housing presented in section 6.2.2 this thread is not used, but for an aluminum replacement for the ball nut housing it could be utilized.

An aluminum part consisting of a block and a cylinder could be machined. The proposed design is shown in Fig. 6.18. The top cylinder would be machined to fit precisely into the hole in the coordinate table axis, and the block portion of the part would have a hole in the side of it for the ball screw. The hole shown in the figure is threaded part of the way with a thread matching the one on the ball screw. This is where the ball screw would be mounted. In this way, the ball screw would no longer be contained within two halves of a part, but would screw into the side of the part connecting it to the axis.

The advantage of this design is that it could be achieved with 2.5D machining G-code, whereas if the previous 3D printed design would require 3D machining G-code if the cylinder shape of the inside hollow were to be machined. The design also contains just a single part, instead of two separate halves. The downside is that for both the top cylinder and the side hole and tap to be machined, the workpiece would have to be remounted midway on the built three-axis CNC milling machine. This can be done, it just takes a bit longer to finish the part than if all machining operations could be done without remounting the workpiece.

### 6.9.4 Spindle mounting plate

The spindle mounting plate currently used with the machine is made out of a 20 mm thick slab of aluminum with a hole to mount the spindle in. There is a setscrew for fastening the spindle in the hole. There is also a smaller hole beside the spindle mounting hole, probably for mounting an optional coolant fluid distribution system to. The size of the spindle mounting plate is 120 mm × 70 mm, within the limits of what the milling machine can make. As a coolant fluid system is not installed on the milling machine, a proposed improved version of the mounting plate could have a mounting hole for a vacuum cleaner hose instead of the small hole currently available. Mounting a vacuum cleaner hose right beside the milling bit would be useful, as cutting chips could then be removed while the milling was in progress without having to manually hold the vacuum cleaner up to the work.

## 6.10 Self-improvement

This chapter has documented the improvements made to the first iteration of the CNC milling machine, and has been an extension of process of building the machine. In this chapter some parts created for the machine by the machine itself have been detailed, and this shows that the machine is capable of creating self-improving by creating parts for itself. The milling of PCBs shows that the machine is capable of good precision, and the milling of the work holding clamps shows that the machine is capable of creating parts out of aluminum.

In addition to the milled components detailed in the chapter, the last part of the chapter introduced some more components that it would have been possible to let the machine create for itself. Together the components described in this chapter provide an answer to the second problem posed in the introduction of the thesis: "Which components of a CNC machine are the ones which are most suitable for improvement?" The components discussed can be considered suitable due to their size not exceeding the CNC machine's working area, and for being specifically designed to be easy to create on a 3-axis milling machine.

# Chapter 7

# Resulting Machine

This chapter presents a short overview of the results achieved in the process of building the CNC milling machine together with the process of improving the machine. It will show how the finished CNC milling machine turned out, including technical specifications, cost of materials and performance of the machine. Some aspects of the built machine will be compared with the other previously detailed CNC machines.

## 7.1 Machine specifications

The completed CNC milling machine is shown in Fig. 7.1. There is a view of the back of the machine in Fig. 7.2. The machine is pictured with a machine vise clamped to the working table. Rubber damping feet have been mounted to the bottom of the frame, raising the machine up off from the surface it is placed on and performing as vibration isolation. This helps to reduce the operating noise of the machine.

The different stepper motor mounts have been labeled with the name of the axis as well as arrows indicating the positive and negative directions of motion. The peculiar signal yellow color of the Z-axis stepper motor mount has no function, but is merely a result of the 3D printer running out of the white plastic building material at the time of printing. It is not easily seen in Fig. 7.1, but small clear pieces of polycarbonate sheet have been taped to the tops of the X- and Y-axis stepper motor mounts. This has been done to keep chips away from the bearings and couplings housed inside.

The ball screw of the Y-axis may currently be exposed to metal chips during milling operations. It is desirable that chips are kept away from the ball screw. This is a precision component, and the precision could be affected if chips are allowed to accumulate on it. Some simple chip guards could be made out of thin plastic sheet folded into an accordion shape and mounted to cover the exposed ball screws. This will be done as soon as some suitable chip guard material is located, but until that time the machine works well without it.

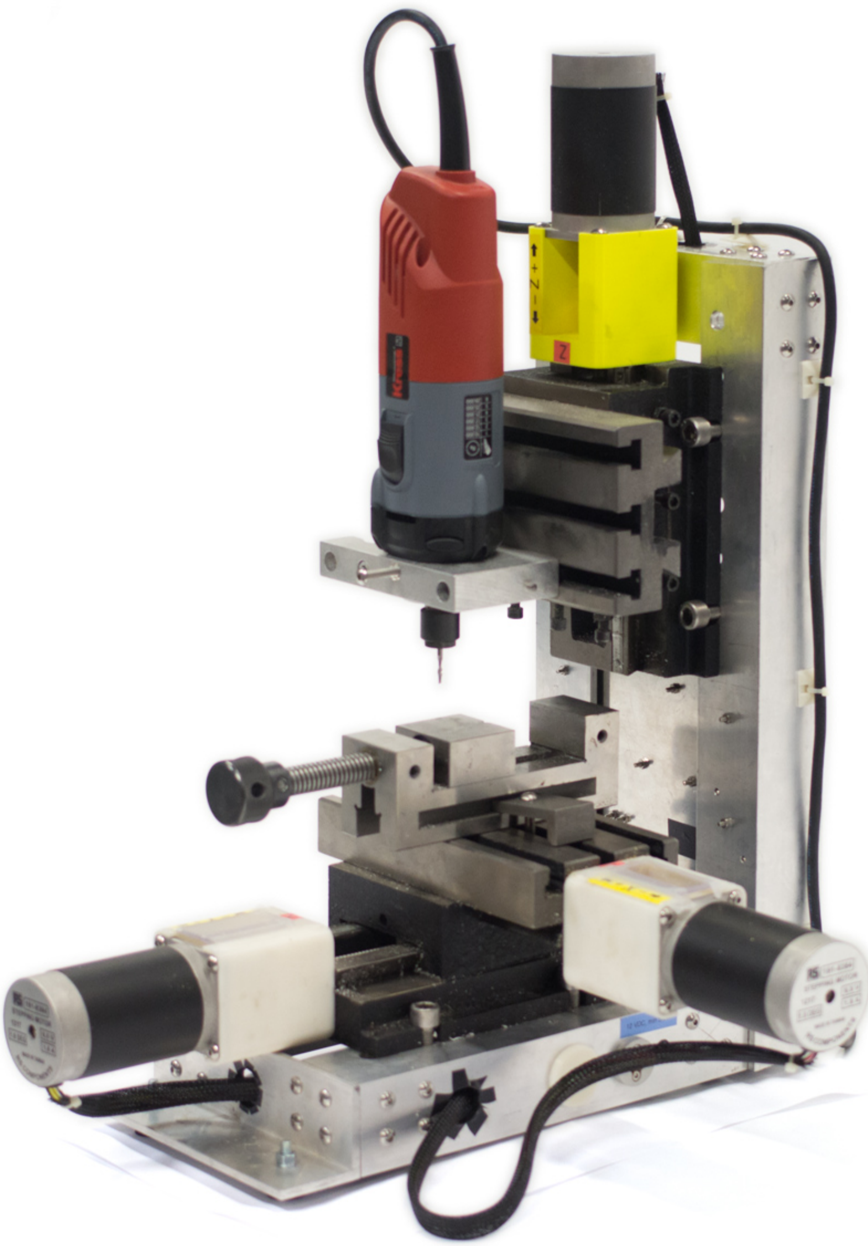


**Fig. 7.1** The finished CNC milling machine, as built.

To use the machine a power cable and a USB communications cable must be connected. The connection sockets for these are mounted on the bottom horizontal of the frame, but the view of them is partly obstructed by the black cable coming off the X-axis stepper motor. The USB connector has been relocated to this position for ease of access. It is easier to connect a USB cable to the side of the machine than to connect it into the Arduino located at the back of the machine. When operating the machine an additional separate spindle power cable must be plugged in. The spindle is operated manually by turning it off and on with the power switch and adjusting the speed of it with an adjustment wheel. This works well when using the machine.

The electronics wiring has been done as neatly as possible, and cable ties have been used to keep everything organized. The wires of the stepper motors have been bunched together by using black cable sleeves and have been routed along the inside the aluminum frame of the machine to keep chips away. The holes in the frame through which the stepper motor wiring is run have been lined with tape so the rough edges of the holes do not wear on the wires. When extending the stepper motor wires to reach all the way to the electronics boards, the same wire colors as used on the stepper motors themselves have been used. Heat shrink tubing has been used throughout the assembly to avoid exposed bare metal wires.

All input and output terminals on the custom created PCBs and the Arduino board have been labeled with signal names and the color corresponding to the wire color used for that signal. The PCBs have been labeled with descriptions of what the function of the board is. Keeping the electronics wiring neat and labeled provides documentation of the work and makes it easier for future modifications or repairs to be made. The electronics is designed to be as modular as possible, using screw terminals to ensure dependable electrical connections.

The back view in Fig. 7.2 reveals that the X-axis stepper motor protrudes from the rest of the machine by a fair amount. The weight of the stepper motor may act as a lever on the worktable of the machine and could potentially cause the table to be not entirely square with respect to the spindle. This is remedied by ensuring that the tightening screws of the X-axis adjustment gib are tightened correctly. The gib and its function was discussed in more detail in section 6.3. When the gib adjustment screws are tight enough, the tilt is taken out of the X-axis. The Y-axis does not experience the same issue, as that stepper motor is connected to the part of the coordinate table which is bolted directly to the frame.

An overview of the technical specifications for the completed CNC milling machine is shown in Table 7.1.

The machine is quite heavy, as the two cast iron coordinate tables used in the machine weigh 11 kg each. The machine is probably heavier than all of the four hobbyist CNC machines discussed in section 2.4. A heavy machine may be a nuisance when moving it, but is actually favorable for a machine when removing material from a workpiece. A heavier machine is more rigid

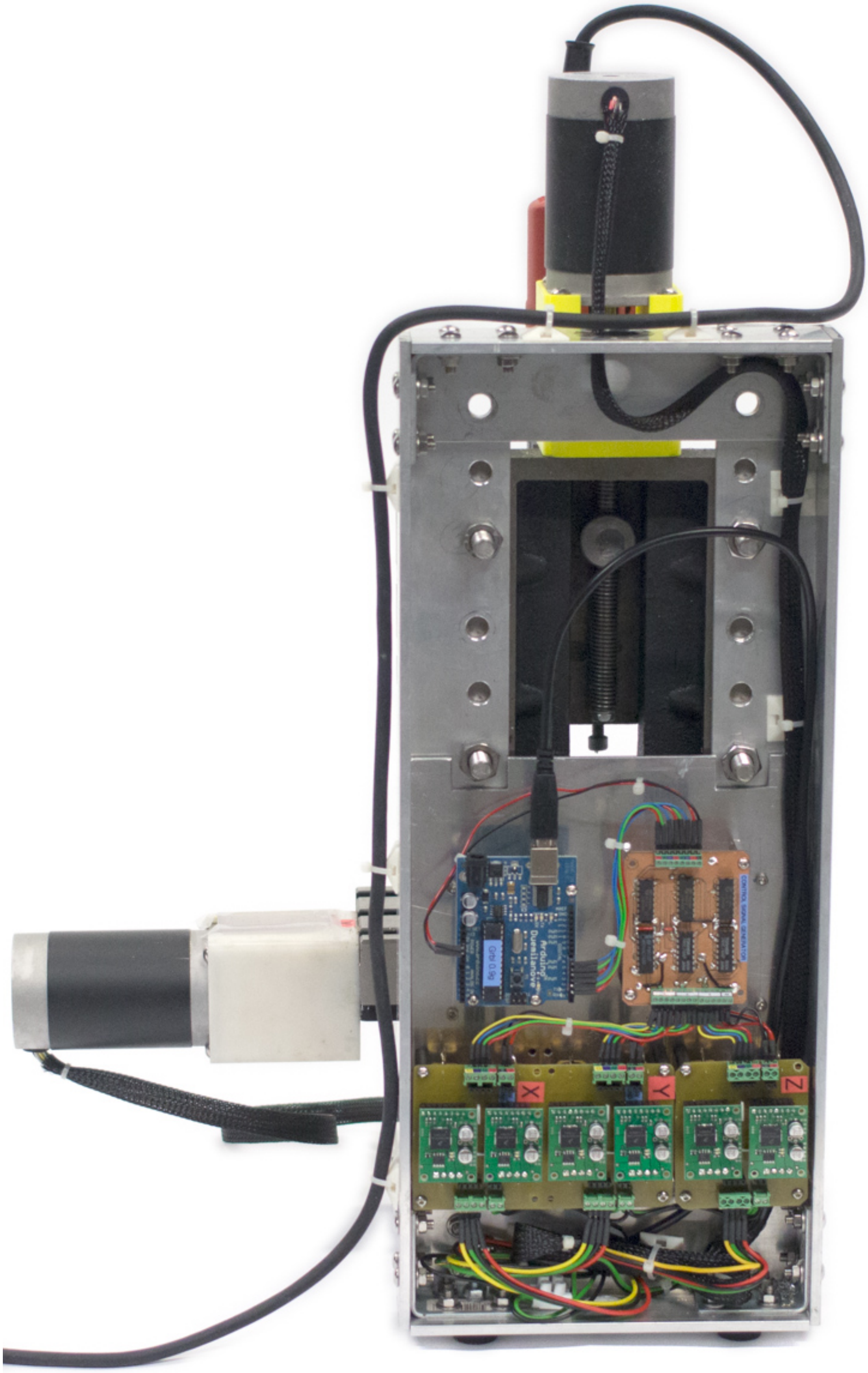



**Fig. 7.2** Back of the finished CNC milling machine, showing the electronics assembly mounted in place.

**Table 7.1** Built CNC milling machine specifications

| | Built CNC machine |
|---|---|
| Stepper motors | NEMA 23 |
| Spindle | Kress 800 FME,<br>420 W,<br>10 000–29 000 rev/min |
| Dimensions (mm) | 450 × 450 × 560<br>(at full extension) |
| Working area (mm) | 130 × 74 × 74 |
| Max. speed (mm/min) | 150 |

than lighter machines, as the increased weight serves to resist the forces encountered when doing heavy cuts in a workpiece.

Compared to the four CNC milling machines detailed in section 2.4, the built machine uses larger and heavier stepper motors. The bigger stepper motors mean that the built machine probably has a higher torque available when milling, which combined with the powerful spindle allows materials to be milled as well as or better than the other machines. The stepper motors of the Creation Station seem to be around the same size with comparable torque to what is delivered by the stepper motors of the built CNC milling machine.

The spindle of the built machine is a bigger and better spindle than the Dremel or Dremel-like tools used with the DIY LIL CNC and the Shapeoko. The construction of the built CNC milling machine also is sturdy enough to easily support a spindle of this size and weight. The power of the spindle of the built CNC milling machine is comparable to the one used with the Lobo CNC milling machine with its 440 W. The most powerful spindle is the one used with the Creation Station, which is rated at 750 W.

Although the built CNC milling machine is heavy and has powerful stepper motors, the physical footprint of the machine is smaller than most of the other CNC machines it has been compared with. The Lobo CNC machine seems to be about the same size. The small size of the built CNC machine means that the working area is smaller than the other machines, but the smaller size probably means that it is also a more sturdy machine, and is therefore able to take on tougher working materials.

The built CNC milling machine has a larger working area than the other two scratch-built CNC mills that have been created at the research group, the BabyMill and the MidiMill. Those two smaller CNC mills have been built primarily as educational tools to help teach students about CNC machines. They are not required to be able to mill tough materials, as long as the machine axes move in the way specified by the G-code fed to them. The built CNC milling machine has had a slightly different design focus, and is therefore capable of milling a larger variety of materials.

Although the speed of all of the different comparison CNC machines is not

**Table 7.2** Cost of the main components

| Part name | Part no. | Qty. | Total cost (NOK) | Supplier |
|---|---|---|---|---|
| Coordinate table 105 × 190 mm | HBM01144 | 2 | 2 580 | www.gasolin.no |
| NEMA23 stepper motor | 191-8384 | 3 | 1 652 | no.rs-online.com |
| SKF 8 mm ball screw shaft w/ nut | 341-1762 | 1 | 898 | no.rs-online.com |
| SKF cylindrical ball nut for 8 mm shaft | 481-2655 | 1 | 602 | no.rs-online.com |
| Ruland 6 mm shaft collar | 449-0564 | 6 | 191 | no.rs-online.com |
| Ruland 6 mm shaft jaw coupling | 423-1696 | 3 | 266 | no.rs-online.com |
| Alu. angle 50 × 50 × 5 mm (L: 100 mm) | 446-6729 | 2 | 213 | no.rs-online.com |
| Kress 800 FME spindle | | 1 | 1 000 | www.damencnc.com |
| MC33887 motor driver carrier board | 711 | 6 | 648 | www.pololu.com |
| *Subtotal:* | | | *8 050* | |

known exactly, it seems like a speed of 2000 mm/min is not unobtainable. The Creation Station CNC machine advertises a speed of 2000 mm/min on its website, and the Shapeoko website conservatively suggests a default seek speed of 500 mm/min. The built machine has a speed which is nowhere near this figure, with its 150 mm/min. This speed is a bit too slow and means that milling parts takes a long time. The speed can probably be increased quite a bit by tweaking the Grbl settings and making sure that the movement of all of the axes is as smooth as possible. Increasing the speed of the built CNC machine can therefore probably be performed with no further expense to the machine.

## 7.2 Cost of materials

The cost of the main components of the built CNC milling machine is summarized in Table 7.2. The table gives an indication of the cost of the machine. The cost of the minor components like the nuts and bolts of the machine have been left out of the summary. The labor that has gone into the machine is similarly not registered as this would yield a very high cost which would not be easily comparable to the other CNC machines discussed in section 2.4. When comparing the cost of the produced machine to the other machines which were summarized in Table 2.1, the built CNC machine seems to come out a bit expensive. However, when remembering that a purchase of one of the existing CNC machines would entail a shipping fee and an added import tax (into Norway) of 25 %, the built CNC machine is around the middle of the scale.

During the building of the machine more focus has gone into using components and materials already available at the research group than trying to keep the cost of the machine down. This may have driven the cost of the machine up slightly.

## 7.3 Performance of the machine

The built CNC milling machine performs well and has been able to create a number of parts. The ball screws installed, detailed in section 6.2.2, ensure that the precision of the machine is such that it can create usable parts without visible backlash problems.

The machine has been designed so that it can be built with few specialized tools. The Lobo CNC mill requires a specialized CNC folding machine to create its custom structural parts, something which may put off some hobbyist users who would have liked to build these components from scratch.

The built CNC machine comes out on top when comparing to the Creation Station with respect to delivery date. The built machine has been finished and is ready to start creating parts, whereas the Creation Station is still not being shipped to end users, and there are indications that shipping internationally may not be offered at all.

In all, the built machine is a compact and powerful CNC machine capable of milling in various materials including aluminum. The milling strength of the built machine probably compares favorably with machines such as the Shapeoko or the DIY LIL CNC, which seem like they would struggle with milling aluminum without extensive modifications. Some of the objects created with the built CNC machine are shown in section C of the appendix.

## 7.4 Self-improvement results

The machine has performed some self-improvement, in that it has milled out printed circuit boards for its own electronics. This is an important upgrade from the prototype setup with temporarily wired breadboard electronics, as it in practice means that the machine may be moved around easily. It also means that the machine is robust enough to survive being used at the research group.

Workholding clamps were the first thing to be created on the machine when starting out with milling in aluminum. A total of four aluminum workholding clamps have been made. The clamps have been used by the machine on subsequent milling jobs to hold flat workpieces down securely, so milling accessories like this for the machine represents a self-improvement by the built CNC milling machine.

A set of aluminum parallels were also created by the machine. These have been used by the machine itself when using the machine vise to hold workpieces. The parallels lift the workpiece up from the bottom of the vise, preventing damage to the vise from the milling bit digging into the material of it.

# Chapter 8

# Discussion

In this chapter a general discussion of the thesis work is presented. Some of the individual components of the machine and why they were used are explained. A conclusion is given, summarizing the work done concerning the thesis. The end of the chapter gives an overview of proposed future work that could be performed on the built CNC milling machine to further increase its usefulness.

## 8.1 General discussion

The built CNC milling machine still has 3D printed parts. The most noticeable plastic parts are the stepper motor holders, but there are also 3D printed plastic parts in the Z-axis custom shaft coupling and the X- and Y-axis ball nut housings. In the introduction chapter of the thesis it was stated that a goal for the machine was to have the machine create as many parts for itself as possible. The reason why the built machine still has a number of 3D printed plastic parts is that these parts perform their intended functions well, and therefore focus was shifted towards other parts of the machine when these parts had been created. Alternative designs have been proposed for these components. These designs were presented in section 6.9.

### 8.1.1 Coordinate tables

Using a coordinate table for the basis of a CNC milling machine was a nice way of getting started building something quickly. As the coordinate table is designed for machining use, it takes little effort to motorize the two axes to achieve X- and Y-axis movement in a basic way. Getting the Z-axis in place is a bit more difficult, as a frame or support structure needs to be designed. Using a coordinate table for the Z-axis works well, but a lot of unnecessary weight is added, as only one of the two moving axes is needed, and the other is just fixed in place.

An alternative solution would be to keep one coordinate table in place for the X- and Y-axes, but create a custom carriage for the Z-axis. The Z-axis carriage plate could be fitted with ball bearings that would roll on two parallel aluminum angle lengths mounted vertically on the frame. This would cut down on the weight of the final CNC milling machine, and would cut the cost of the machine, as only one coordinate table would be necessary.

### 8.1.2 Ball screws

While building the CNC milling machine it quickly became apparent that the backlash in the coordinate tables was unacceptable, and that this would have to be compensated for somehow. While an early idea was to compensate for the backlash in software, it quickly became apparent that the preferred way to deal with the backlash was to use a mechanical solution. The reason for this was that software compensation would have prevented the G-code to be executed as smoothly as it could be with a mechanical backlash reduction solution.

Ball screws were used, as these are utilized in commercial CNC machines. Some effort went into converting the CNC milling machine from using the existing Acme screw to using ball screws, but the result was that the backlash was practically eliminated.

### 8.1.3 Motor mounts

The motor mounts have not been remade in aluminum because the 3D printed plastic versions function well in the current milling machine. Since the 3D printed motor mount is made as a single block, and not assembled out of different parts, it is rigid enough to handle the torque applied by the stepper motors.

The alternative motor mounts presented in section 6.9 could have been created, as the built CNC milling machine is capable of making the parts needed for this. The advantage of rebuilding the motor mounts out of aluminum would have been that these would possibly be more durable than the 3D printed plastic ones. In addition, the alternative design guards the enclosed ball bearings and the shaft coupling from milling chips, something which was not thought of when the plastic motor mounts were designed.

### 8.1.4 Custom shaft coupling

The alternative threaded coupling hub design has not been tested out on the built CNC milling machine. The 3D printed plastic custom hub still does its job, but if it eventually wears out, the alternative threaded coupling hub may be tested.

## 8.2 Conclusion

This thesis has detailed the work in creating a CNC milling machine for the ROBIN research group at the University of Oslo. The machine has been designed and built to a working standard. It is ready for use when manufacturing small parts.

The intent of the CNC milling machine was to create as many parts for the machine as possible on the machine itself, but this has only been done with a few parts, as the other parts made out of plastic work well. The machine has manufactured four workholding clamps, its own PCB electronics boards using the previously manufactured workholding clamps, and a small power connector holder. The small power connector holder was held in a machine vise where it was elevated with self-created parallel blocks.

The different PCBs were the main priority to machine on the CNC milling machine itself, as these were parts that could not be 3D printed, and as such would need to be machined. A total of four PCBs were created on the CNC milling machine, one of these double sided. The PCBs for the permanent electronics assembly were important upgrades, as this allows the CNC milling machine to be portable. The PCBs were milled after first milling the four workholding clamps.

One of the specific problems posed in the introductory chapter of this thesis was: "How can a self-improving CNC milling machine be created?" Given that the CNC milling machine built during the work with this thesis has been able to manufacture its own PCBs, the machine is one way of creating such a self-improving CNC milling machine. It has created parts which it utilizes to create other parts. The machine has also created a small power connector holder part, mainly as a proof of concept, but in combination with the four work holding clamps and the set of parallels, it shows that the machine is able to manufacture parts from aluminum, opening up for further machine part manufacturing.

New aluminum parts to substitute the stepper motor mounts, the Z-axis custom coupling, and the ball nut holders used on the X- and Y-axes have not been manufactured on the CNC milling machine. Designs for these upgraded parts have been proposed. As the machine is able to machine aluminum, and the proposed upgraded parts are not bigger than what the machine can handle, these parts could also be created on the machine.

This also answers the second problem posed in the introduction: "Which components of a CNC machine are the ones which are most suitable for improvement?" Of the mentioned parts, the ones that are most suitable for improvement are the smallest parts, as these require less machining time. The necessity for turning the workpiece and refitting it to the worktable or in the vise is also a factor, as this significantly increases machining time. PCBs are also easily created on the built CNC milling machine, as long as the size of the PCB does not exceed the working are of the machine.

The least suitable components for self-improvement are components that are larger than the working area of the CNC machine. To overcome the working area limitation, an indexing system could be utilized where indexing marks could be created on the part while milling a certain area, so that the workpiece could be precisely located with the help of the indexing marks when milling a different area of the part.

Other components unsuitable for self-improvement are parts that need to be created out of a material that the built CNC milling machine is not able to easily manufacture parts out of. Different kinds of steel have not been tested out with the built CNC milling machine, so parts that need to be made out of steel are less suitable for self-improvement.

## 8.3 Future work

Although the CNC milling machine built during the work with this thesis is now able to successfully manufacture small parts, there are some areas of improvement that may be explored to create an even better machine.

### 8.3.1 Motor speed

The built CNC milling machine performs well and creates good parts, but even so the machine could be improved by increasing the operation speed of the machine. In that case the speed of the individual stepper motors would have to be increased. In the current machine all of the individual axes have a maximum speed of 150 mm/min. When milling aluminum this is not too bad, but it would be preferable to have a faster speed for traveling and positioning the cutter. It would also be nice to have when milling softer materials like foam.

Judging from other hobby CNC machines online, it should be possible to get more speed out of a setup run by Grbl, as discussed at the end of section 7.1. Having a faster speed may lead to the machine getting more use just from being more efficient and therefore more enjoyable to use. The CNC milling machine built during the work with this thesis is limited to 150 mm/min in the Grbl setup, as this was the speed at which the stepper motors would run before missing steps and becoming erratic. Increasing the speed may be as easy as trying out different Grbl configuration values and ensuring that the different axes run as smoothly as possible to avoid any excessive friction which may hinder faster speeds.

Trying to increase the speed of the machine is potentially time consuming, but the benefits of having a faster machine would easily justify the time spent on the attempt. This would be a welcome improvement if anyone would try to perform the modification.

### 8.3.2 Increasing the number of axes

Many milling operations can be done with three axes of movement, but if a part with an overhang is to be manufactured, one or two additional axes may be required. Even though the machine may be too small to be able to convert it from a 3-axis machine to a 5-axis machine, it may be feasible to build a detachable fourth axis controlled by a stepper motor that could be mounted to the work table. This would be a nice addition to the machine, for example when cutting gears.

### 8.3.3 Servo motor control

Stepper motors have been used to build the CNC milling machine of this thesis. This has been advantageous as stepper motors are inexpensive compared to a servo motor solution. It would however be interesting to see how the milling machine would perform with servo motors instead of stepper motors driving the axes. It would also be possible to experiment with running stepper motors as servos like the Lobo CNC milling machine discussed in section 2.4.4. If successful, this would eliminate any inaccuracy problems due to lost stepper motor steps. It would also be able to provide the operator with a digital read-out of his current position.

### 8.3.4 Increased working area

A larger working area would be a useful modification to the machine, as the working area limits the size of parts that can be comfortably created on the machine. A design could be proposed where the range of the coordinate tables used in the built CNC milling machine is extended by bolting on extra sections. Together with replacing the existing lead screws with longer ones, this could extend the range of the axis. A lot could be achieved with just increasing the length of the X-axis. Many manual milling machines have X-axes with much longer travel than the Y-axis.

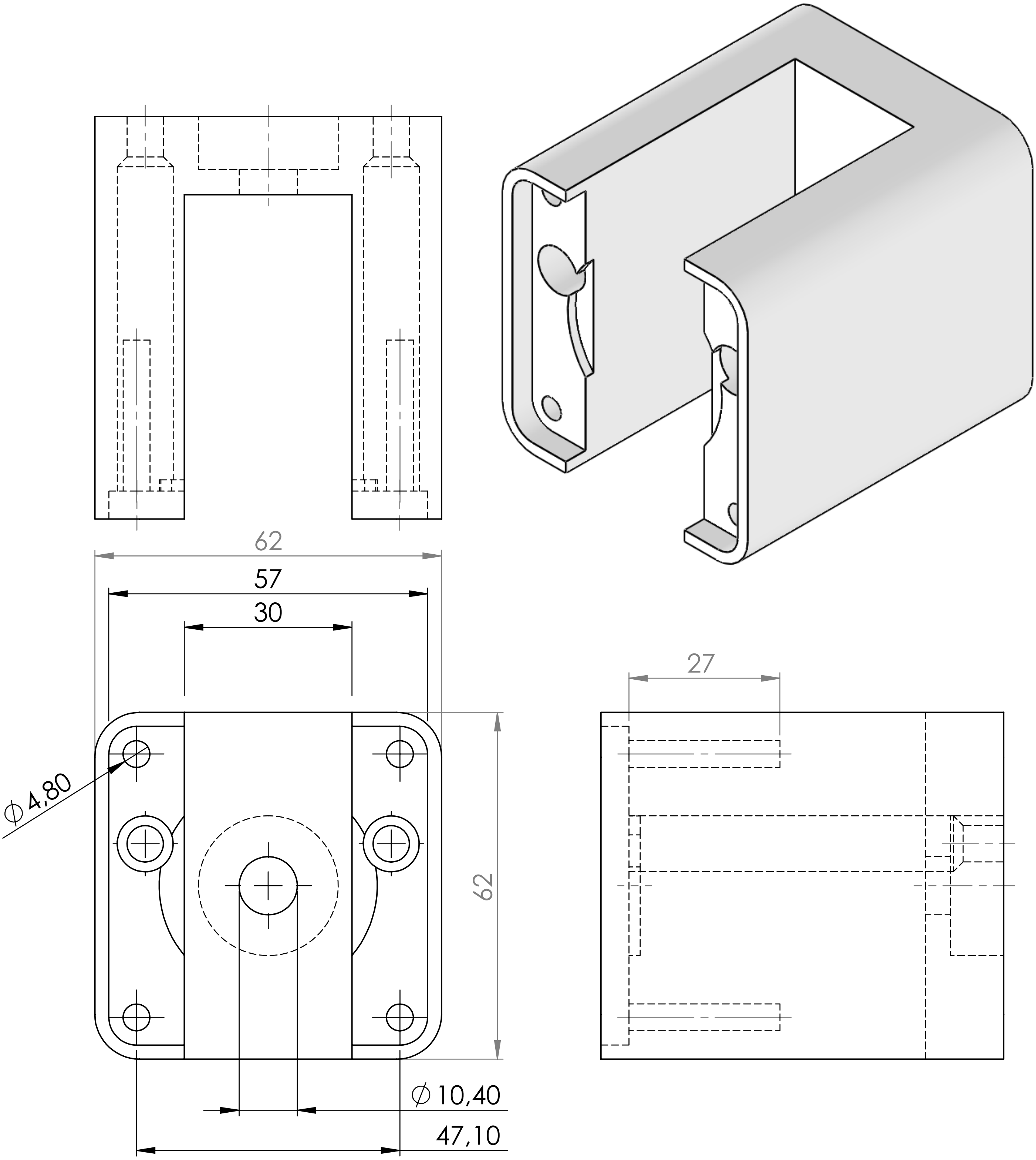
## 8.4 Summary

The work with this thesis has provided the author with an opportunity to build a CNC machine. The hands-on mechanical work has been enjoyable and the process has been a learning curve as many aspects of CNC machine design and metalworking have been explored. As a functional CNC machine has been completed, the author hopes that the machine will see further use, hopefully becoming a valuable addition to the manufacturing equipment available at the robotics and intelligent systems research group. The author further hopes that the members of the research group feel free to further modify the built CNC milling machine if this leads to further improvements of the machine.

# Appendix A

# Technical Drawings

This part of the appendix includes mechanical drawings of the custom parts that have been designed and created in the process of building the CNC milling machine. All drawings are created by the author.



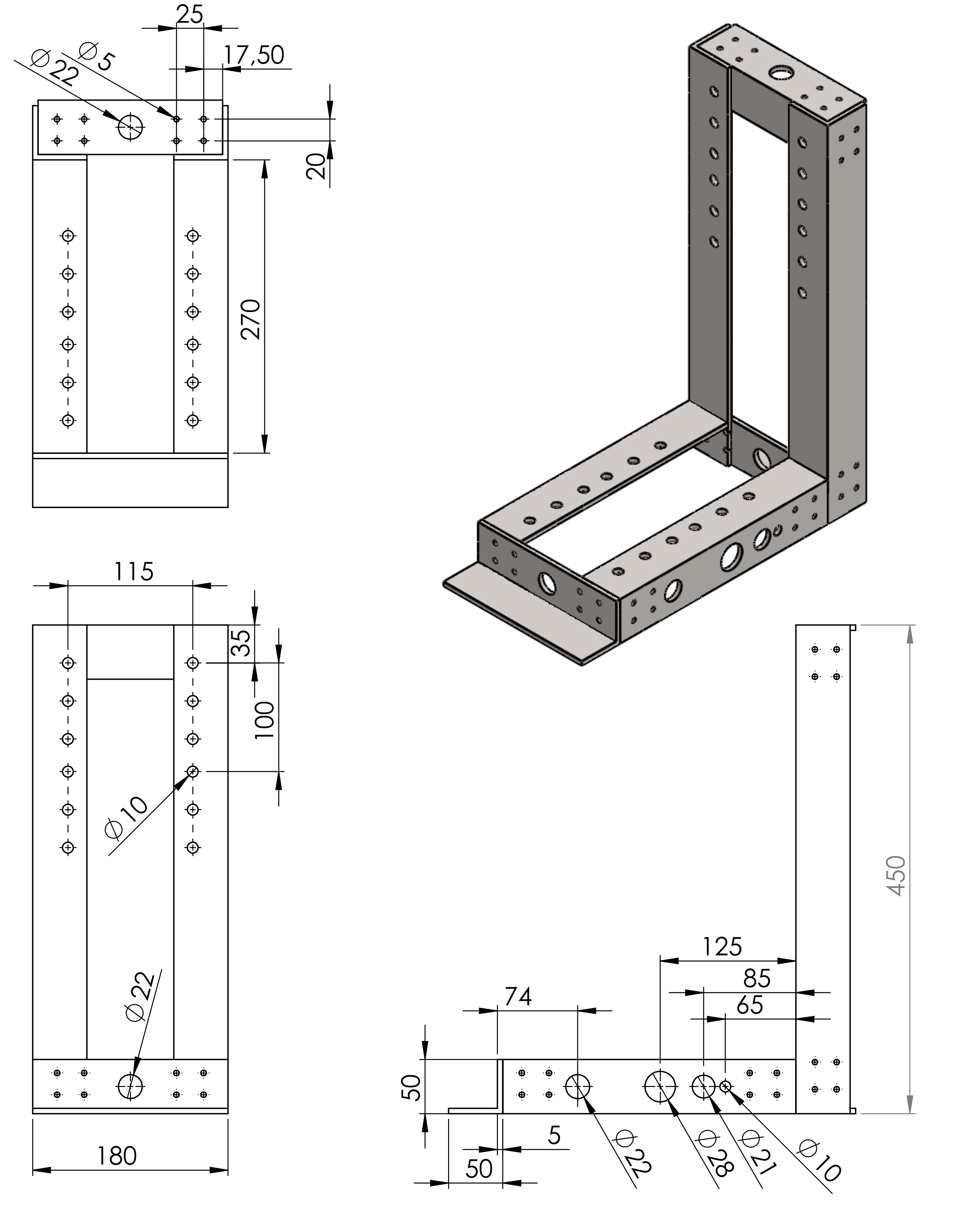

| | | | | |
|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR: | FINISH: | DEBUR AND BREAK SHARP EDGES | DO NOT SCALE DRAWING | REVISION |

| | NAME | SIGNATURE | DATE | |
|---|---|---|---|---|
| DRAWN | torjuss | | 31.01.2014 | TITLE: |
| CHK'D | | | | |
| APPV'D | | | | |
| MFG | | | | |
| Q.A | | | | MATERIAL: ABS plus plastic |

DWG NO. **motor_mount** A4

WEIGHT:

SCALE:1:1 SHEET 1 OF 1




UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | torjuss | | 27.11.2014 |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:
Aluminum

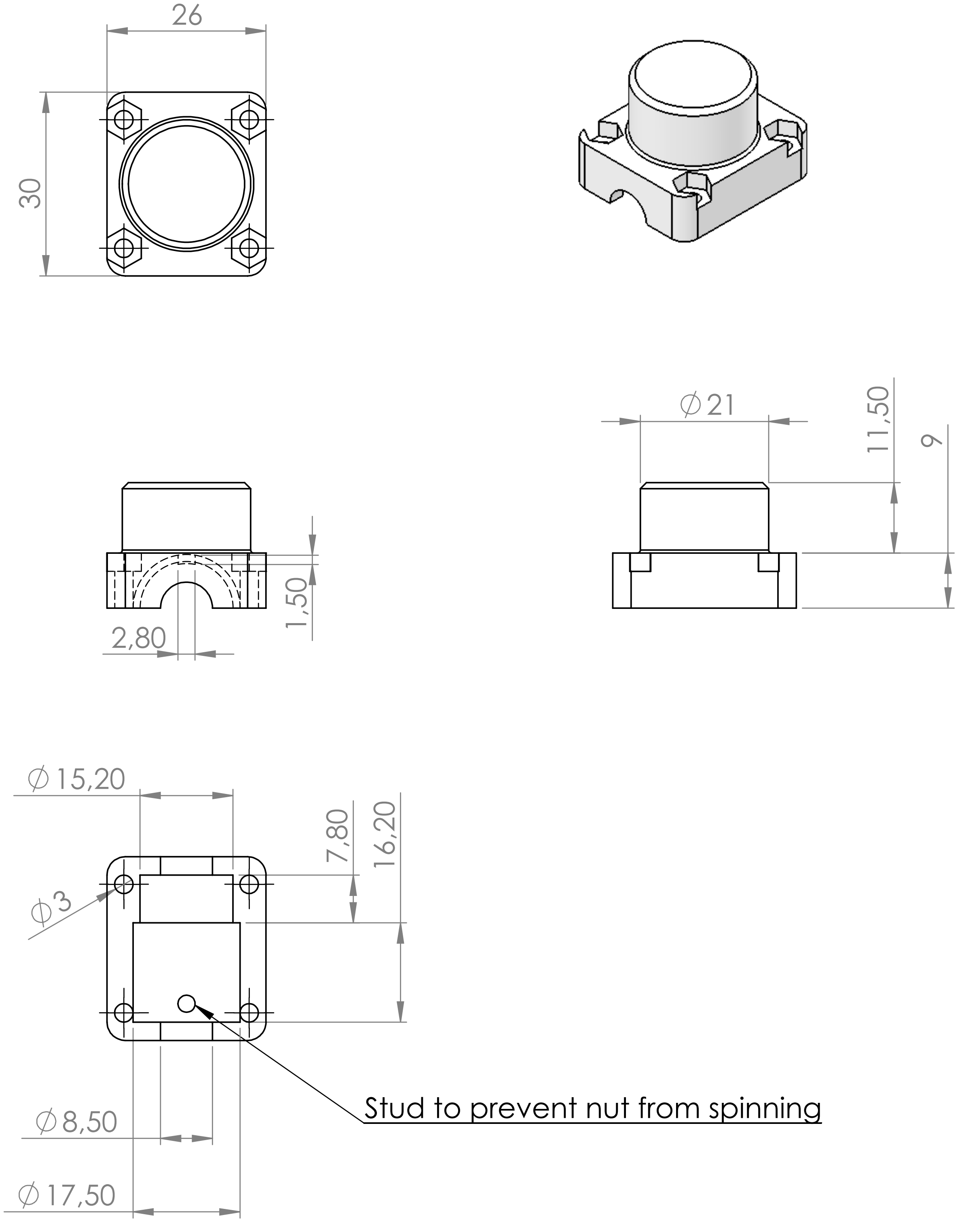
WEIGHT:

DWG NO.
frame

A4

SCALE:1:5

SHEET 1 OF 1




UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | torjuss | | 6.10.2014 |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:
ABS plus plastic

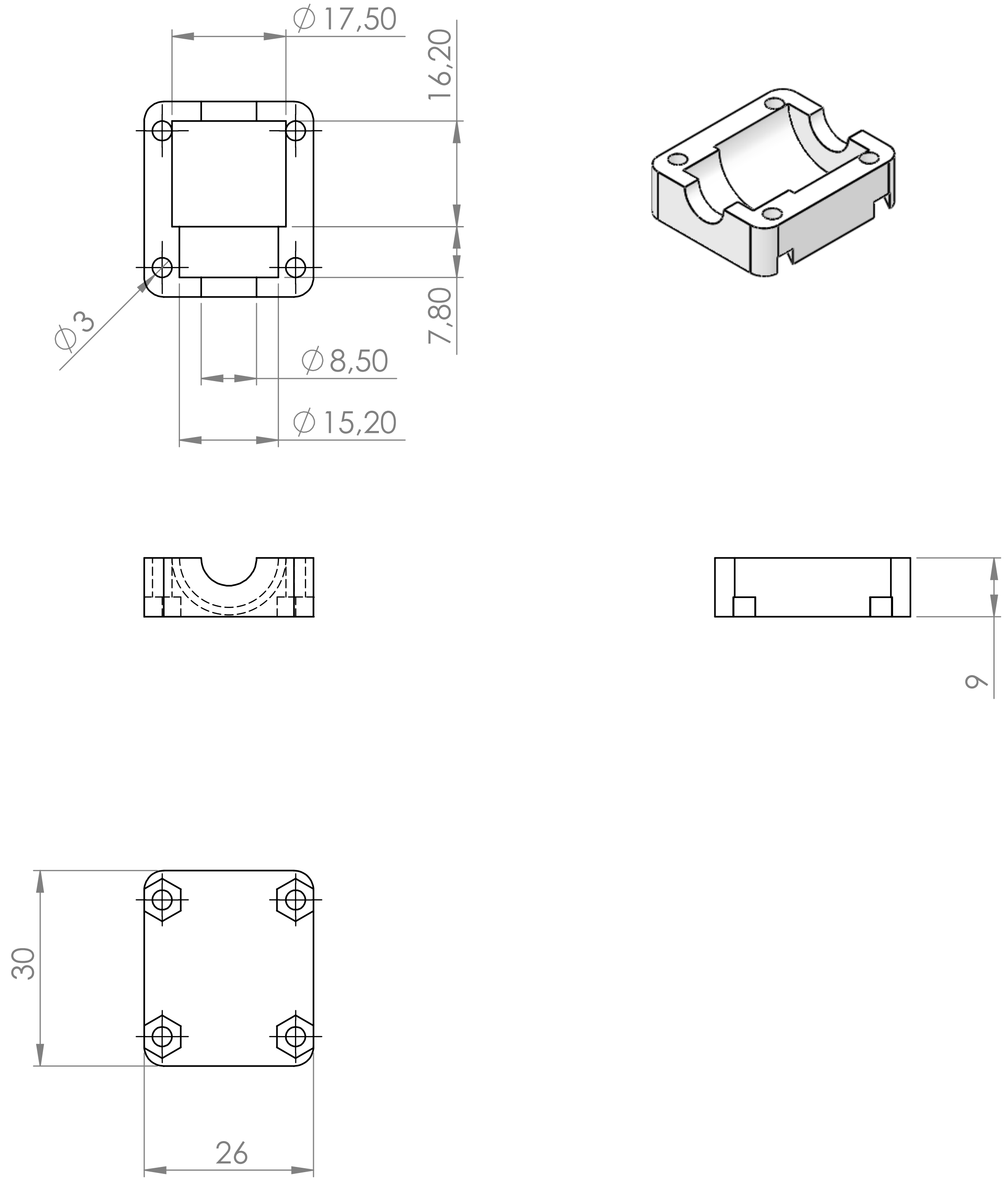
WEIGHT:

DWG NO.
nut_holder_top

A4

SCALE:1:1

SHEET 1 OF 1




| | | | | | | |
|---|---|---|---|---|---|---|
| UNLESS OTHERWISE SPECIFIED:<br>DIMENSIONS ARE IN MILLIMETERS<br>SURFACE FINISH:<br>TOLERANCES:<br>LINEAR:<br>ANGULAR: | FINISH: | | DEBUR AND<br>BREAK SHARP<br>EDGES | DO NOT SCALE DRAWING | REVISION | |
| | NAME | SIGNATURE | DATE | TITLE: | | |
| DRAWN | torjuss | | 6.10.2014 | | | |
| CHK'D | | | | | | |
| APPV'D | | | | | | |
| MFG | | | | | | |
| Q.A | | | MATERIAL:<br>ABS plus plastic | DWG NO.<br>nut_holder_bottom | A4 | |
| | | | WEIGHT: | SCALE:1:1 | SHEET 1 OF 1 | |

# Appendix B

# Electronics

This part of the appendix includes the schematics for the electronic circuits designed for the CNC milling machine. Also included are the PCB designs for these electronic circuits.

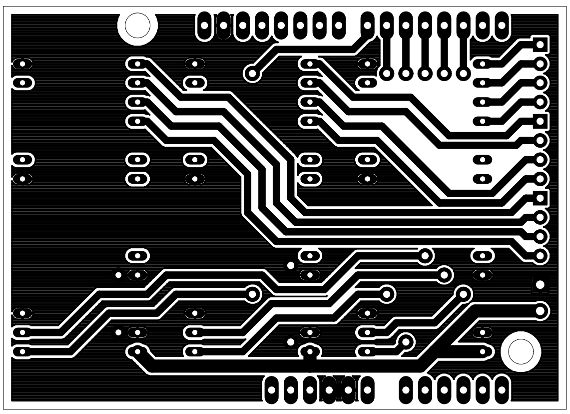


**Fig. B.1** PCB for EasyDriver socket shield for Arduino. Used with BabyMill and MidiMill.

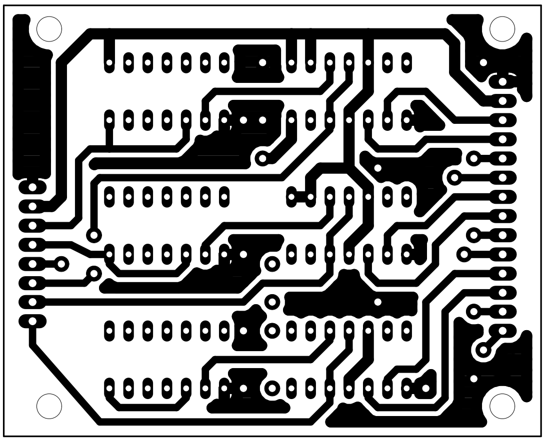


**Fig. B.2** Step motor control signal generator PCB. Bottom side.

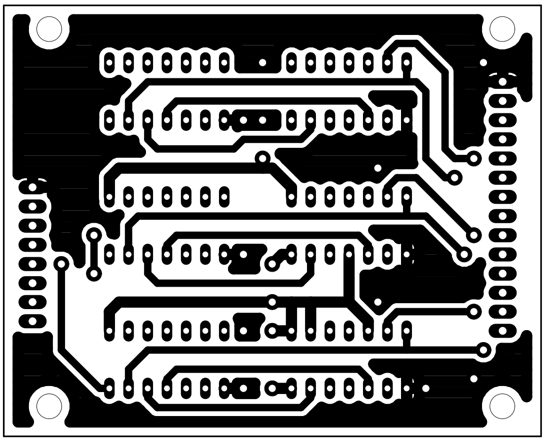


**Fig. B.3** Step motor control signal generator PCB. Top side.

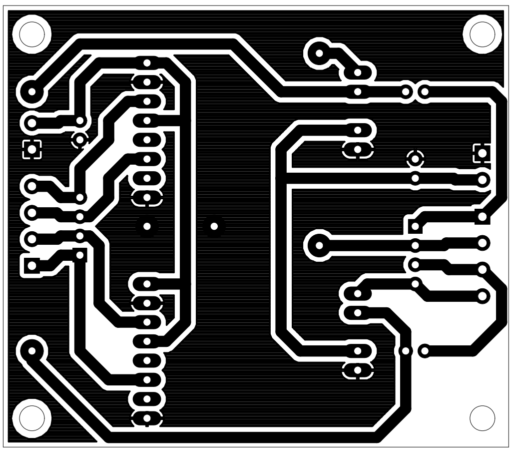
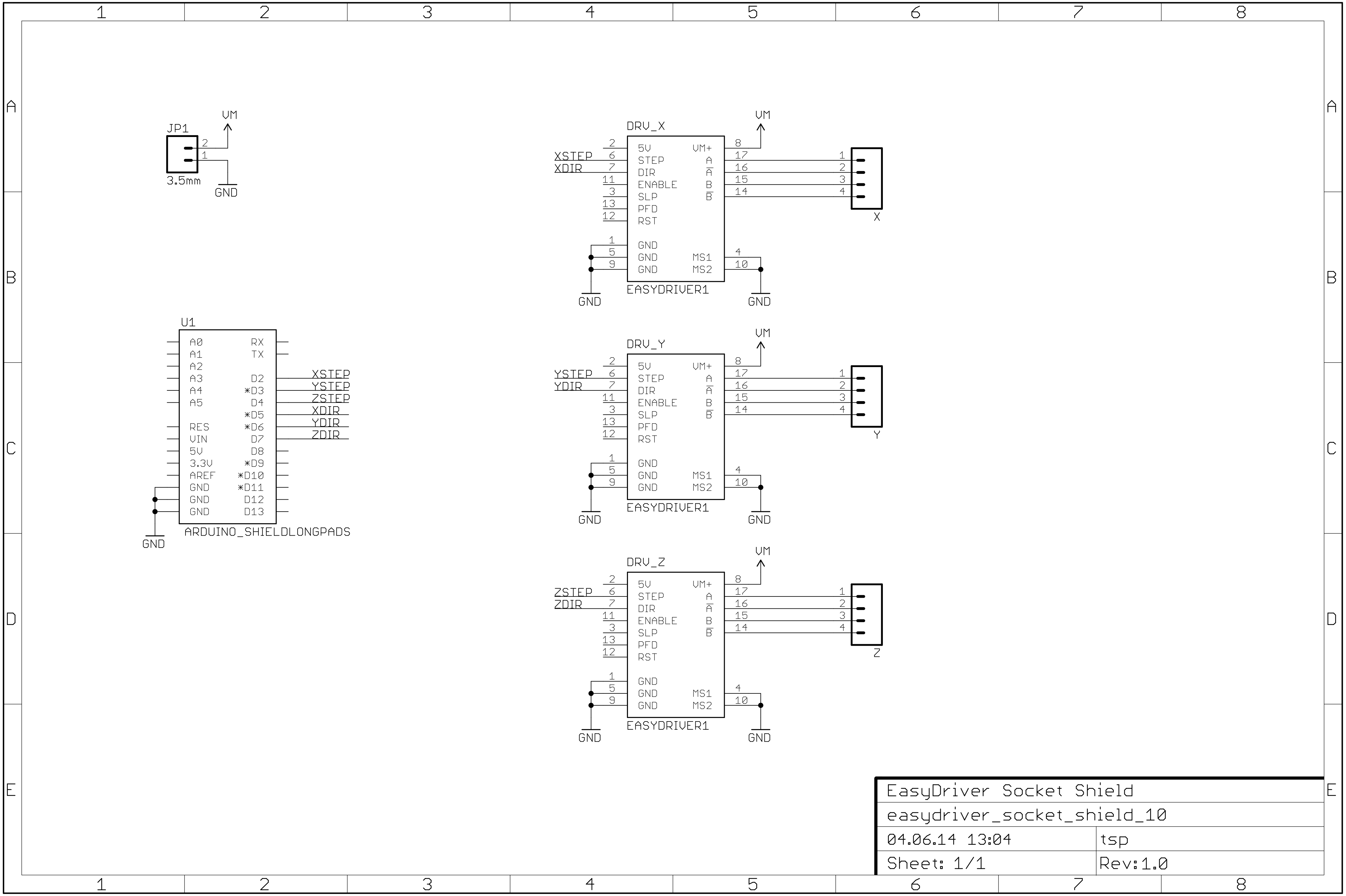
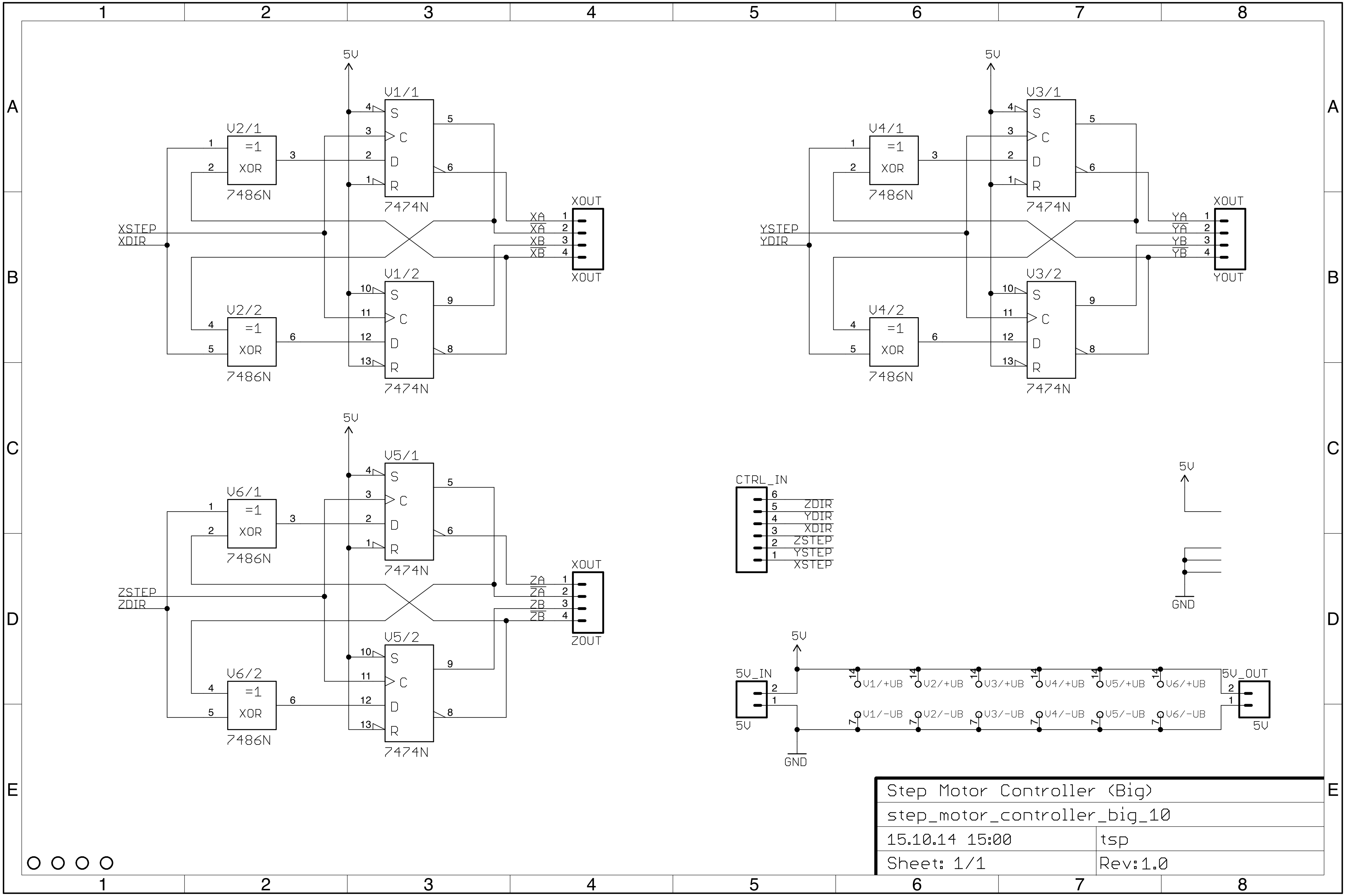
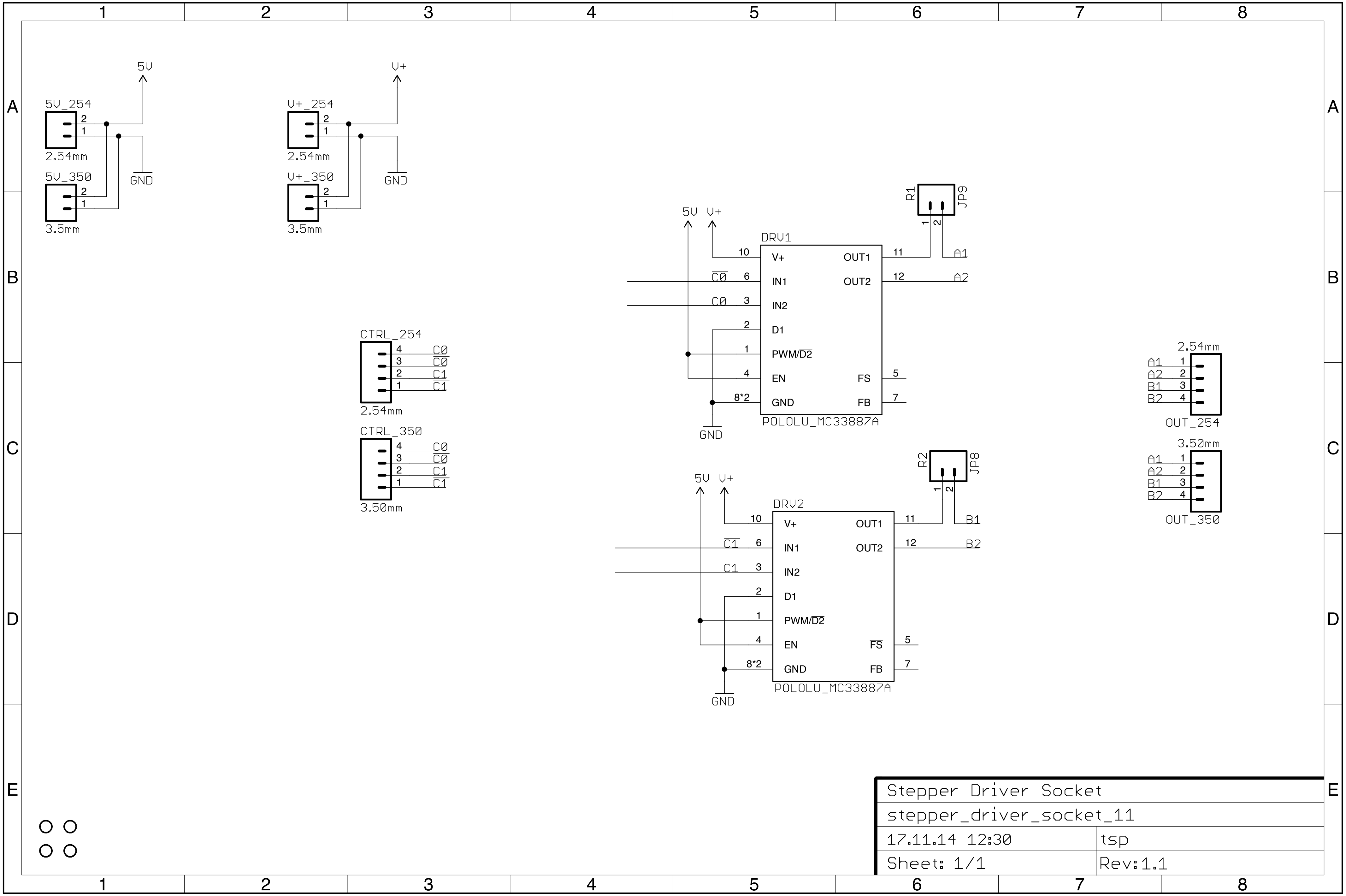


**Fig. B.4** Stepper driver socket PCB. Three of these were created and used with the big CNC milling machine.



JP1
VM
GND
3.5mm
U1
A0 A1 A2 A3 A4 A5 RES VIN 5V 3.3V AREF GND GND GND
RX TX D2 *D3 D4 *D5 *D6 D7 D8 *D9 *D10 *D11 D12 D13
XSTEP YSTEP ZSTEP XDIR YDIR ZDIR
ARDUINO_SHIELDLONGPADS
DRV_X
DRV_Y
DRV_Z
5V STEP DIR ENABLE SLP PFD RST GND GND GND
VM+ A A B B MS1 MS2
EASYDRIVER1
X
Y
Z
EasyDriver Socket Shield
easydriver_socket_shield_10
04.06.14 13:04
tsp
Sheet: 1/1
Rev:1.0


Step Motor Controller (Big)
step_motor_controller_big_10
15.10.14 15:00
tsp
Sheet: 1/1
Rev:1.0
U1/1
U1/2
U2/1
U2/2
U3/1
U3/2
U4/1
U4/2
U5/1
U5/2
U6/1
U6/2
7474N
7486N
XOR
=1
XSTEP
XDIR
YSTEP
YDIR
ZSTEP
ZDIR
XOUT
YOUT
ZOUT
XA
XB
YA
YB
ZA
ZB
CTRL_IN
5V_IN
5V_OUT
5V
GND


5V_254
2.54mm
5V_350
3.5mm
5V
GND
V+_254
2.54mm
V+_350
3.5mm
V+
GND
CTRL_254
2.54mm
CTRL_350
3.50mm
C0 C0 C1 C1
DRV1
POLOLU_MC33887A
V+ IN1 IN2 D1 PWM/D2 EN GND
OUT1 OUT2 FS FB
R1 JP9
A1 A2
DRV2
POLOLU_MC33887A
R2 JP8
B1 B2
OUT_254
2.54mm
OUT_350
3.50mm
A1 A2 B1 B2
Stepper Driver Socket
stepper_driver_socket_11
17.11.14 12:30
tsp
Sheet: 1/1
Rev:1.1

# Appendix C

# Milled Objects

This section shows some of the objects that have been created with the built CNC milling machine.

**Fig. C.1** Front half of a power connector holder.


**Fig. C.2** Sheriff badge. Milled out of 1 mm brass plate. Engraved by a 0.2 mm 45° engraving bit.

**Fig. C.3** A set of workholding clamps. Body is milled out of aluminum.

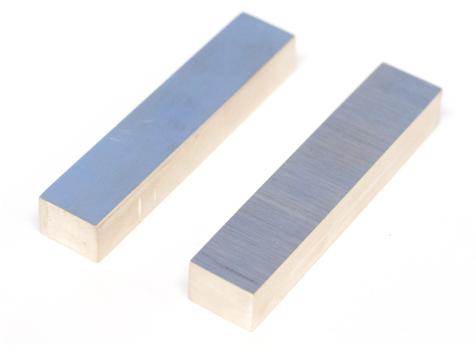


**Fig. C.4** A set of parallels. Used with a machine vise to lift the workpiece off the bottom to prevent milling into the vise. The black T-nuts at the end of the threaded rods connect with T-slots in the worktable.

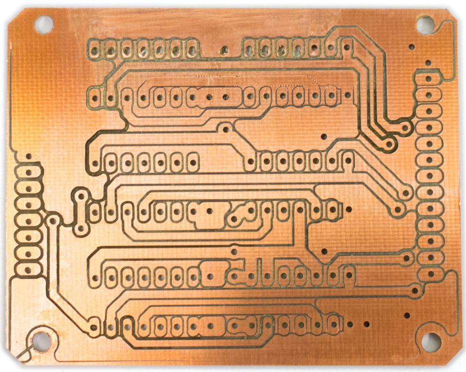


**Fig. C.5** Step motor control signal generator PCB.

# Bibliography

*Arduino website* (2005). URL: www.arduino.cc (visited on 2014-11-15).

Black, Roman (2008). *Good CNC designs*. URL: www.romanblack.com/cnc_good.htm (visited on 2014-09-25).

Bowyer, Adrian (2014). "3D printing and humanity's first imperfect replicator." In: *3D printing and additive manufacturing* 1.1, pp. 4–5. URL: online.liebertpub.com/toc/3dp/1/1.

Colvin, Fred Herbert and Daniel John Duffin (1947). *60 years with men and machines: an autobiography*. New York: McGraw-Hill.

Dring, Barton (2011). *MakerSlide website*. URL: www.makerslide.com (visited on 2014-09-25).

Freescale Semiconductor (2012). *MC33887 datasheet. 5.0 A H-bridge with load current feedback*. Rev. 16.0. URL: www.freescale.com/files/analog/doc/data_sheet/MC33887.pdf.

*Grbl Wiki* (n.d.). URL: www.github.com/grbl/grbl/wiki (visited on 2014-11-15).

Hokanson, Taylor and Chris Reilly (2009). *DIYLILCNC website*. URL: www.diylilcnc.org (visited on 2014-11-15).

Hokanson, Taylor and Chris Reilly (2011). "DIYLILCNC: open source plans for a low-cost desktop CNC mill." In: *ACM SIGGRAPH 2011 studio talks*. ACM, p. 8.

Hood-Daniel, Patrick and James Floyd Kelly (2009). *Build your own CNC machine*. Berkeley, CA: Apress. DOI: 10.1007/978-1-4302-2490-7.

Innovation Squared LLC (2014). *The Creation Station open source CNC router Kickstarter page*. URL: www.kickstarter.com/projects/1417197574/the-creation-station-open-source-cnc-router (visited on 2014-11-17).

*Instructables website* (n.d.). URL: www.instructables.com (visited on 2014-11-15).

*ISO website* (n.d.). URL: www.iso.org (visited on 2014-11-22).

Jones, Rhys et al. (2011). "RepRap – the replicating rapid prototyper." In: *Robotica* 29.Special Issue 01, pp. 177–191. DOI: 10.1017/S026357471000069X. URL: http://dx.doi.org/10.1017/S026357471000069X.

Kerr, Jeff (2013). *Building aluminum guitars: the open-source Lobo CNC*. URL: www.makezine.com/2013/06/19/building-aluminum-guitars-the-open-source-lobo-cnc/.

Kerr, Jeff (n.d.). *Lobo CNC website*. URL: www.jrkerr.com/lobocnc (visited on 2014-11-15).

Lipson, Hod and Melba Kurman (2013). *Fabricated: the new world of 3D printing*. Indianapolis, IN: Wiley.

Lombard, Matt (2013). *Solidworks 2013 bible*. Indianapolis, IN: Wiley.

*MakerCAM website* (n.d.). URL: www.makercam.com (visited on 2014-09-25).

Mota, Catarina (n.d.). "The rise of personal fabrication." In: *Proceedings of the 8th ACM conference on creativity and cognition*. ACM, pp. 279–288.

Neumann, John von (1966). *Theory of self-reproducing automata*. Urbana, IL: University of Illinois Press.

oomlout (2007). *How to make a three axis CNC machine (cheaply and easily)*. URL: www.instructables.com/id/How-to-Make-a-Three-Axis-CNC-Machine-Cheaply-and-/ (visited on 2014-11-14).

Philips Semiconductors (1990). *74HC/HCT86 datasheet. Quad 2-input exclusive-OR gate*.

*Pololu Robotics and Electronics website* (n.d.). URL: www.pololu.com (visited on 2014-11-15).

*Project Shapeoko website* (2011). URL: www.shapeoko.com (visited on 2014-09-25).

*Shapeoko Wiki* (2011). URL: www.shapeoko.com/wiki/ (visited on 2014-11-15).

Skogsrud, Simen Svale (n.d.). *Bengler: Grbl website*. URL: www.bengler.no/grbl (visited on 2014-11-15).

*SparkFun Electronics website* (n.d.). URL: www.sparkfun.com (visited on 2014-11-15).

Stallman, Richard (2007). *GNU general public licence v3*. Free Software Foundation. URL: www.gnu.org/licenses/gpl.html.

Stroud, Ian and Hildegarde Nagy (2011). *Solid modelling and CAD systems: how to survive a CAD system*. London: Springer. DOI: 10.1007/978-0-85729-259-9.

Sutherland, John W. (2002). *CYBERMAN: an educational manufacturing page*. URL: www.mfg.mtu.edu/cyberman/ (visited on 2014-11-19).

*The Creation Station CNC Router website* (n.d.). URL: www.creationstationcnc.com (visited on 2014-11-15).

Toshiba (2007). *74HC74AP datasheet. Dual D-type flip-flop with preset and clear*.

Trigg, Len and Richard White (2013). *NoLash: software backlash compensation*. URL: www.github.com/Lenbok/NoLash (visited on 2014-11-18).

Zeid, Ibrahim (2005). *Mastering CAD / CAM*. Boston, MA: McGraw-Hill Higher Education.